

9.4

Supervisión y rendimiento para IBM MQ

IBM

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información en [“Avisos” en la página 395](#).

Esta edición se aplica a la versión 9 release 4 de IBM® MQ y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que considere adecuada, sin incurrir por ello en ninguna obligación con el remitente.

© **Copyright International Business Machines Corporation 2007, 2024.**

Contenido

- Supervisión y rendimiento..... 5**
 - Supervisión de la red de IBM MQ..... 5
 - Integración de OpenTelemetry..... 5
 - Comportamiento de comprobación de estado del gestor de colas..... 6
 - Supervisión de sucesos..... 9
 - Supervisión de mensajes..... 60
 - Mensajes de contabilidad y estadísticas..... 140
 - Rastreo de actividad de la aplicación..... 207
 - Temas del sistema para la supervisión y el rastreo de actividades..... 295
 - Supervisión en tiempo real..... 306
 - Supervisión de clústeres..... 318
 - Supervisión del equilibrio de aplicaciones..... 321
 - Monitoring performance and resource usage on z/OS..... 324
 - Ajuste de la red de IBM MQ..... 380
 - Ajuste canales de conexión cliente y servidor..... 380
 - Ajuste de redes de publicación/suscripción distribuidas..... 382
 - Reducción del número de temas no deseados en el árbol de temas..... 392
 - Aspera gateway puede mejorar el rendimiento en redes de alta latencia..... 394

- Avisos..... 395**
 - Información acerca de las interfaces de programación..... 396
 - Marcas registradas..... 397

Supervisión y rendimiento de IBM MQ

Utilice la información de supervisión y de guía de esta sección y los consejos de ajuste específicos como ayuda para mejorar el rendimiento de la red del gestor de colas.

Acerca de esta tarea

Dependiendo del tamaño y la complejidad de la red del gestor de colas, puede obtener información diversa como resultado de la supervisión de la red. Puede utilizar esta información, junto con la información que se proporciona en consejos de ajuste específicos, para ayudarle a ajustar el rendimiento de la red.

Supervisión de la red de IBM MQ

IBM MQ proporciona varias técnicas de supervisión para obtener estadísticas y otra información específica sobre el funcionamiento de la red del gestor de colas. Utilice la información de supervisión y de guía de esta sección como ayuda para mejorar el rendimiento de la red del gestor de colas.

Estas son algunas razones para supervisar la red del gestor de colas:

- Detectar problemas en la red del gestor de colas.
- Ayudar en la determinación de las causas de problemas de la red del gestor de colas.
- Mejorar la eficacia de la red del gestor de colas.
- Familiarizarse con la ejecución de la red del gestor de colas.
- Comprobar que la red del gestor de colas funciona correctamente.
- Producir mensajes cuando se producen determinados sucesos.
- Registrar la actividad de mensajes.
- Determinar la última ubicación conocida de un mensaje.
- Comprobar diversos datos estadísticos de una red del gestor de colas en tiempo real.
- Generar un seguimiento de auditoría.
- Determinar cómo se utilizan los recursos de aplicación.
- Planificar la capacidad.

▶ V 9.4.0

▶ ALW

Integración de OpenTelemetry

Puede integrar IBM MQ con un sistema de rastreo de OpenTelemetry .

Rastreo de OpenTelemetry

El rastreo de OpenTelemetry le permite observar cómo se comportan las aplicaciones dentro de un flujo de datos. El flujo de datos puede, y a menudo lo hace, incorporar varias aplicaciones diferentes. El rastreo puede mostrar todo el recorrido y proporcionar información sobre el comportamiento de cada aplicación individual. IBM MQ proporciona un servicio de rastreo que le permite integrar con un sistema de rastreo de OpenTelemetry .

El servicio de rastreo de IBM MQ OpenTelemetry se implementa como una salida de API de IBM MQ . Se proporciona como un programa de soporte de IBM , lo que significa que tiene titularidad para utilizar el servicio de rastreo y recibir soporte como parte de la titularidad de IBM MQ . Tenga en cuenta que la titularidad de IBM MQ sólo permite el uso del componente de salida de rastreo de IBM MQ del programa de soporte de IBM Instana .

Los problemas derivados del uso de la salida deben notificarse al soporte de IBM MQ a menos que esté utilizando la salida de rastreo con un sistema de supervisión de IBM Instana y una titularidad de IBM Instana .

La salida se puede descargar aquí: <https://ibm.biz/mqinstanaexit>.

Instalación y configuración del servicio de rastreo de IBM MQ Open Telemetry

Aquí puede encontrar todos los detalles sobre cómo instalar y configurar el rastreo de IBM MQ : [IBM MQ Tracing](#).

Los detalles de las plataformas soportadas para la salida de IBM Instana IBM MQ se pueden encontrar aquí: [Plataformas soportadas para IBM MQlocal](#).

Multi

Comportamiento de comprobación de estado del gestor de colas

El gestor de colas realiza comprobaciones de estado periódicas para garantizar un rendimiento estable y fiable. En este tema se describen algunas de las comprobaciones de estado que realiza el gestor de colas y se explica cómo se pueden configurar en función de los requisitos del entorno.

En la mayoría de los entornos, la configuración predeterminada es adecuada y no es necesario cambiar la frecuencia con la que se realizan estas comprobaciones. Incluso utilizando los valores predeterminados, puede ser útil comprender cómo se comporta el gestor de colas cuando se detecta un problema y los problemas ambientales que pueden hacer que falle una comprobación. Este tema está pensado para explicar algunos de estos comportamientos.



PRECAUCIÓN: No realice ningún cambio en la frecuencia de estas comprobaciones a menos que se lo indique el servicio de soporte de IBM .

Los distintos componentes del gestor de colas utilizan diversos medios para detectar y resolver incoherencias y este tema no está pensado para describir todos estos mecanismos. Por ejemplo, los procesos de IBM MQ utilizan diversos mecanismos para asegurarse de que otros procesos de los que dependen siguen en ejecución. Los comportamientos que se describen son aquellos comportamientos que realiza periódicamente el controlador de ejecución para detectar situaciones ambientales u otras inesperadas. (El controlador de ejecución es el proceso primario de IBM MQ que inicia y gestiona la mayoría de los demás procesos del gestor de colas.) Puesto que son comprobaciones periódicas, se realizan en intervalos específicos que se pueden modificar hasta cierto punto estableciendo los parámetros de ajuste adecuados.

Algunas de las comprobaciones descritas las realiza una hebra de comprobación de estado dedicada. Si se detecta un problema con la propia hebra de comprobación de estado, se graba un mensaje de aviso [AMQ5066](#) en los registros de errores del gestor de colas.

Los comportamientos que se describen en este tema están sujetos a cambios en futuros releases, por ejemplo, si se observa que un valor predeterminado diferente es más estable en una plataforma o configuración determinada.

Controles sanitarios generales

El gestor de colas realiza diversas comprobaciones en un intervalo regular. De forma predeterminada, estas comprobaciones se realizan cada 10 segundos (en algunos casos, la comprobación permite dos ciclos antes de informar de un error, lo que da como resultado un intervalo de 20 segundos para dichas comprobaciones). Durante las comprobaciones, el gestor de colas se asegura de que los distintos procesos que se ejecutan como parte del gestor de colas siguen en ejecución. Para un gestor de colas de HA nativa, comprueba que el gestor de colas está replicando correctamente los datos en las instancias en espera.

Si una comprobación crítica falla en este momento (por ejemplo, si el proceso `amqzmc0` ya no está en ejecución), el gestor de colas no podrá continuar ejecutándose. Sin embargo, la mayoría de las comprobaciones se realizan para ordenar los recursos del sistema que ya no son necesarios y pueden simplemente hacer que se graba un mensaje en los registros de errores del gestor de colas.

En la mayoría de los casos, no es necesario cambiar la frecuencia de estos controles sanitarios generales. La mayoría de los sucesos en el gestor de colas o en el entorno operativo se detectan de forma instantánea sin necesidad de que el proceso de comprobación de estado general los detecte. Este proceso sirve como una comprobación periódica de cualquier cosa que no se detecte en otro lugar del gestor de colas. Si es necesario, la frecuencia se puede configurar utilizando el parámetro de ajuste **ECHeartBeatLen**. El valor mínimo es 10000 milisegundos (10 segundos). El valor máximo es 60000 milisegundos (60 segundos). Si se establece en su valor máximo de 60000, esto puede dar como resultado un retardo de dos minutos para determinadas comprobaciones.

Comprueba que se está realizando el progreso del registro

Las comprobaciones del gestor de colas que graban en el registro se realizan a una velocidad razonable. Esto no es una comprobación de que el rendimiento del registrador sea óptimo, sino que está diseñado para detectar condiciones que pueden necesitar más atención. Por ejemplo, si el disco en el que se almacenan los archivos de registro es especialmente lento, o si el gestor de colas no está recibiendo suficiente tiempo de CPU en un entorno contenerizado para realizar todo su trabajo.

Si esta comprobación falla, la acción realizada por el gestor de colas depende del tipo de gestor de colas que se utilice:

- En un gestor de colas no HA:
 - Se escribe un FDC `xecl_W_PERFORMANCE_BOTTLENECK`. Esto se puede utilizar como una indicación de que alguna parte del sistema puede necesitar más atención. El gestor de colas permanece en ejecución. Si los FDC `xecl_W_PERFORMANCE_BOTTLENECK` se ven en el directorio de errores, es posible que sea necesario trabajar con el equipo de almacenamiento o plataforma para comprender si los recursos del sistema subyacentes son suficientes para que IBM MQ se ejecute. Si IBM MQ se está ejecutando en contenedores en nodos sobrecomprometidos, es posible que IBM MQ no reciba suficiente tiempo de CPU planificado para realizar toda su carga de trabajo de mensajería.
 - A partir de IBM MQ 9.3.0, se graba un mensaje de aviso `AMQ5068W` en los registros de errores del gestor de colas y no se graba ningún FDC `xecl_W_PERFORMANCE_BOTTLENECK`. Si los mensajes de `AMQ5068W` se ven en los registros, es posible que sea necesario trabajar con el equipo de almacenamiento o plataforma para comprender si los recursos del sistema subyacentes son suficientes para que se ejecute IBM MQ. Si IBM MQ se está ejecutando en contenedores en nodos sobrecomprometidos, es posible que IBM MQ no reciba suficiente tiempo de CPU planificado para realizar toda su carga de trabajo de mensajería. Si se escriben cinco mensajes de aviso `AMQ5068W` de forma sucesiva, se escribe un FDC `xecl_W_PERFORMANCE_BOTTLENECK`.
- En un gestor de colas de varias instancias:
 - Si la comprobación de estado del progreso del registro falla, la instancia primaria finaliza. Si hay una instancia en espera disponible, se iniciará y se convertirá en la instancia primaria.
 - A partir de IBM MQ 9.3.0, la instancia primaria comprueba si hay una instancia en espera disponible antes de finalizar. Si un gestor de colas en espera está disponible para la migración tras error a la instancia primaria finaliza. Además, se graba un mensaje de aviso `AMQ5068W` en los registros de errores del gestor de colas.
- En un gestor de colas HA nativo, esta comprobación se comporta de la misma forma que un gestor de colas no HA.
- En un RDQM (gestor de colas de datos replicados), esta comprobación se comporta de la misma forma que un gestor de colas no HA.

Es posible que un problema con la progresión del registro de IBM MQ sea causado por un problema de rendimiento en el propio gestor de colas.

De forma predeterminada, esta comprobación se realiza cada 60 segundos, aunque el gestor de colas espera dos ciclos de la comprobación antes de realizar una acción. Esto significa que con los valores predeterminados, tendrían que pasar dos minutos antes de que el gestor de colas escribiera un mensaje de error (o se migrara tras error en el caso de un gestor de colas HA).

En la mayoría de los casos, el comportamiento predeterminado es adecuado incluso cuando el sistema de archivos es lento o cuando al gestor de colas se le asigna una pequeña cantidad de tiempo de CPU porque otras comprobaciones como el bloqueo de archivos (consulte [“Comprueba que los bloqueos de archivo todavía están retenidos”](#) en la página 8) y la operación básica del sistema de archivos provocarán que una instancia primaria realice una migración tras error antes de realizar esta comprobación. Si es necesario, la frecuencia de esta comprobación se puede configurar utilizando el parámetro de ajuste **LivenessHeartBeatLen**. El valor máximo en el que se puede configurar es 600 segundos (10 minutos). El valor mínimo de 0 tiene el efecto de inhabilitar la comprobación por completo. Para un gestor de colas no HA, el único efecto de la comprobación es un mensaje de aviso adicional en los registros de errores del gestor de colas. Para un gestor de colas de varias instancias, puede configurar **LivenessHeartBeatLen** para hacer que una instancia primaria del gestor de colas realice la migración tras error más rápidamente (reduciendo el valor) o lentamente (aumentando el valor). Aumentar el valor para reducir la frecuencia de la comprobación del progreso del registro puede ser útil si el entorno experimenta ocasionalmente una E/S del sistema de archivos muy lenta, pero donde preferiría que la instancia primaria del gestor de colas permaneciera en ejecución. Esto puede ser útil si tiene aplicaciones que no están diseñadas para reconectarse automáticamente a la instancia en espera y requieren intervención manual para reiniciarlas.

Nota: Si se ha aumentado el **ECHearBeatLen**, esto afecta a la temporización de las comprobaciones de **LivenessHeartBeatLen**. Las comprobaciones de progreso de registro se realizan cuando se realizan las comprobaciones de estado generales, por lo que la reducción de la frecuencia de las comprobaciones de estado generales (**ECHearBeatLen**) puede dar como resultado que las comprobaciones de progreso de registro se realicen hasta 30 segundos después del **LivenessHeartBeatLen** configurado.

Rendimiento general del sistema de archivos de registro

▶ V 9.4.0

A partir de IBM MQ 9.4.0, se emite un mensaje de aviso [AMQ6729W](#) en el registro de errores del gestor de colas si las operaciones generales de lectura/grabación en este almacenamiento parecen estar tardando más de lo esperado. Puede utilizar las variables de entorno de **AMQ_IODELAY** para ajustar los diagnósticos y las temporizaciones para ayudar en el diagnóstico de problemas de rendimiento de almacenamiento o para aumentar la tolerancia a dichos retrasos. Para obtener más información, consulte [AMQ_IODELAY](#), [AMQ_IODELAY_INMS](#) y [AMQ_IODELAY_FFST](#).

Comprueba que los bloqueos de archivo todavía están retenidos

Para un gestor de colas de varias instancias, el controlador de ejecución comprueba periódicamente que sigue manteniendo el bloqueo exclusivo en el archivo de varias instancias primario. En muchos casos, si el bloqueo se pierde debido a un problema con el servidor NFS, la instancia primaria migra tras error casi inmediatamente (antes de realizar esta comprobación). Se realizan comprobaciones de bloqueo de archivos periódicas adicionales para asegurarse de que el gestor de colas primario realiza una migración tras error en el caso de un problema inusual del sistema de archivos.

De forma predeterminada, estas comprobaciones de bloqueo de archivo se realizan cada 20 segundos. Si es necesario, este valor se puede cambiar estableciendo el parámetro de ajuste **FileLockHeartBeatLen**. El valor predeterminado para el parámetro de ajuste es 10 segundos (el gestor de colas permite dos ciclos de la comprobación antes de realizar una acción que da como resultado el comportamiento predeterminado de la comprobación cada 20 segundos). El valor mínimo del parámetro de ajuste es de 10 segundos, el valor máximo es de 600 segundos (10 minutos).

Nota: Si **ECHearBeatLen** se ha aumentado, esto afecta a la temporización de las comprobaciones de **FileLockHeartBeatLen**. Las comprobaciones de bloqueo de archivo se realizan cuando se realizan las comprobaciones de estado generales, por lo que la reducción de la frecuencia de las comprobaciones de estado generales (**ECHearBeatLen**) puede dar como resultado que las comprobaciones de bloqueo de archivo se realicen hasta 30 segundos después del **FileLockHeartBeatLen** configurado.

Comprueba el estado de la aplicación de usuario

El gestor de colas comprueba periódicamente que las aplicaciones enlazadas localmente que ya no están en ejecución han realizado una llamada MQI MQDISC antes de terminar. Estas comprobaciones se realizan al mismo tiempo que las comprobaciones de estado generales descritas en [“Controles sanitarios generales”](#) en la [página 6](#). El intervalo predeterminado para estas comprobaciones es, por lo tanto, 10000 milisegundos (10 segundos) y el cambio del valor del parámetro de ajuste **ECHearBeatLen** cambia la frecuencia con la que se realizan. Esta comprobación es principalmente para asegurarse de que los recursos que están asociados con una aplicación conectada se liberen, no hace que un gestor de colas HA o no HA finalice o realice una migración tras error a una instancia alternativa.

Las aplicaciones cliente de IBM MQ que han terminado sin emitir una llamada MQDISC MQI son detectadas por separado por el proceso del agente y se liberan los recursos asociados a la conexión.

Conceptos relacionados

[Configuraciones de alta disponibilidad](#)

Supervisión de sucesos

La supervisión de sucesos es el proceso de detección de las apariciones de los *sucesos de instrumentación* en una red de gestores de colas. Un suceso de instrumentación es una combinación lógica de sucesos detectada por un gestor de colas o instancia de canal. Dicho suceso hace que el gestor de colas o una instancia de canal coloque un mensaje especial, denominado *mensaje de suceso*, en una cola de sucesos.

Los sucesos de instrumentación de IBM MQ proporcionan información sobre errores, advertencias y otras ocurrencias significativas en un gestor de colas. Utilice estos sucesos para supervisar el funcionamiento de los gestores de colas en la red del gestor de colas para conseguir los objetivos siguientes:

- Detectar problemas en la red del gestor de colas.
- Ayudar en la determinación de las causas de problemas de la red del gestor de colas.
- Generar un seguimiento de auditoría.
- Reaccionar ante los cambios de estado del gestor de colas.

Referencia relacionada

[“Tipos de suceso”](#) en la [página 12](#)

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

[Referencia de mensajes de suceso](#)

[Formato de mensaje de suceso](#)

Publicación de mensajes de suceso de IBM MQ

Cómo preparar IBM MQ para publicar mensajes de suceso.

Acerca de esta tarea

Los mensajes de suceso se graban en colas con nombre especial denominadas SYSTEM.ADMIN.<feature name>.EVENT.

Lo importante a tener en cuenta sobre estas colas de sucesos es que el nombre es lo que importa. De forma predeterminada, en un gestor de colas, todas las colas de sucesos se definen como colas locales. Sin embargo, puede suprimir estas colas y volver a definir las, quizás como una cola remota, para que todos los sucesos se canalicen a un gestor de colas de proceso de sucesos dedicado. Como alternativa, puede utilizar una cola de alias que apunte a un objeto de tema.

En cualquiera de los dos casos, cualquier técnica de redirección requiere que las aplicaciones que leen las colas de sucesos no tengan codificado el nombre de la cola que leen. Por lo tanto, debe poder configurar la cola que leen las aplicaciones.

Los mandatos siguientes muestran cómo puede redefinir las colas de sucesos para que se publiquen los mensajes de sucesos, utilizando las presuposiciones siguientes. Ha realizado las siguientes acciones:

- No ha empezado a utilizar sucesos, o
- Ha eliminado todos los mensajes de las colas de sucesos existentes y ha suprimido las colas locales antes de realizar estos pasos.

Estos pasos sólo muestran las colas de sucesos QMGR y CHANNEL que se están redefiniendo, pero esto podría ser ampliable a todos los sucesos.

Nota: La serie de tema está diseñada de forma que una aplicación se pueda suscribir a todos los sucesos utilizando un comodín, o a sucesos específicos, según se requiera.

Procedimiento

Emita los siguientes mandatos:

```
DEFINE TOPIC (ADMIN.QMGR.EVENT) TOPICSTR('Events/QMgr')
DEFINE TOPIC (ADMIN.CHANNEL.EVENT) TOPICSTR('Events/Channel')

DEFINE QALIAS (SYSTEM.ADMIN.QMGR.EVENT) TARGTYPE(TOPIC) TARGET (ADMIN.QMGR.EVENT)
DEFINE QALIAS (SYSTEM.ADMIN.CHANNEL.EVENT) TARGTYPE(TOPIC) TARGET (ADMIN.CHANNEL.EVENT)

DEFINE QLOCAL (ADMIN.EVENT)
DEFINE QLOCAL (ADMIN.QMGR.EVENT)

DEFINE SUB (EVENTS.ALL) TOPICSTR('Events+') PSROP(NONE)
DESTCLAS (PROVIDED) DEST (ADMIN.EVENT)
DEFINE SUB (EVENTS.QMGR) TOPICSTR('Events/QMgr') PSROP(NONE)
DESTCLAS (PROVIDED) DEST (ADMIN.QMGR.EVENT)
```

Suponiendo que la aplicación que lee los sucesos puede leer mensajes de suceso de cualquier cola, esta aplicación se puede reconfigurar desde una de las colas definidas anteriormente según se requiera.

La configuración de `PSROP(NONE)` en los mandatos `DEFINE SUB` es para garantizar que ninguna de las propiedades de mensaje añadidas por el motor de publicación/suscripción, por ejemplo `MQTopicString`, se añade al mensaje de suceso, lo que garantiza que las aplicaciones existentes pueden continuar funcionando sin cambios.

Además, las aplicaciones también se pueden suscribir directamente utilizando la llamada `MQSUB` para recibir la información, como una forma alternativa en lugar de utilizar el mandato administrativo `DEFINE SUB`.

Ahora varias aplicaciones pueden consumir la información emitida en sucesos por el gestor de colas.

Sucesos de instrumentación

Un suceso de instrumentación es una combinación lógica de condiciones que un gestor de colas o una instancia de canal detecta y transfiere un mensaje especial, denominado *mensaje de suceso*, a una cola de sucesos.

Los sucesos de instrumentación de IBM MQ proporcionan información sobre errores, advertencias y otras ocurrencias significativas en un gestor de colas. Puede utilizar estos sucesos para supervisar el funcionamiento de los gestores de colas (con otros métodos como Tivoli NetView para z/OS).

En la [Figura 1](#) en la [página 11](#) se ilustra el concepto de los sucesos de instrumentación.

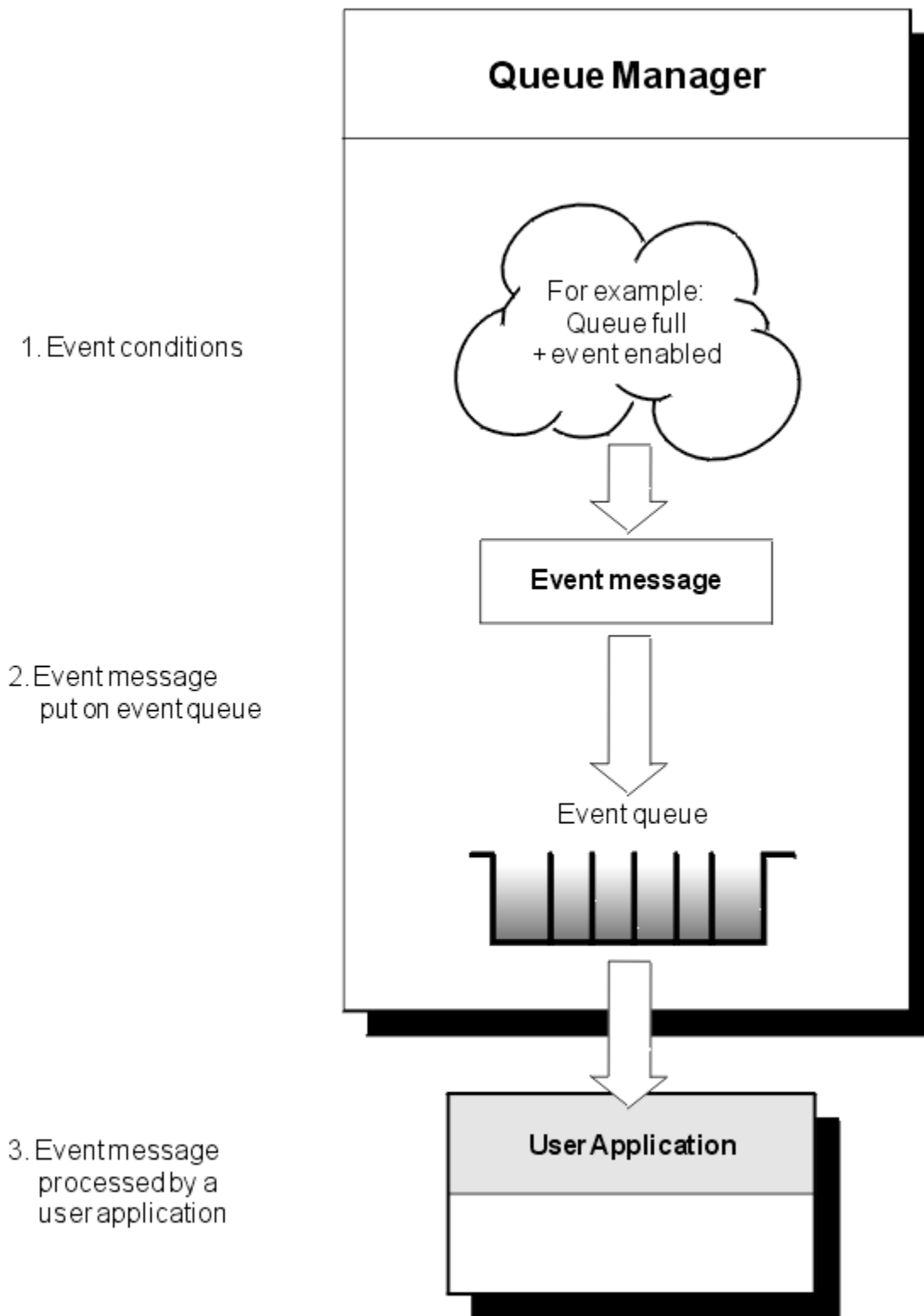


Figura 1. Comprensión de los sucesos de instrumentación

Aplicaciones de supervisión de sucesos

Las aplicaciones que utilizan sucesos para supervisar los gestores de colas deben incluir las siguientes disposiciones:

1. Configurar canales entre los gestores de colas de la red.

2. Implementar las conversiones de datos necesarios. Se aplican las reglas normales de conversión de datos. Por ejemplo, si supervisa sucesos en un gestor de colas del sistema UNIX desde un gestor de colas z/OS, asegúrese de que realiza la conversión de EBCDIC a ASCII.

Notificación de sucesos a través de las colas de sucesos

Cuando se produce un suceso, el gestor de colas transfiere un mensaje de suceso a la cola de sucesos adecuada, si está definida. El mensaje de suceso contiene información acerca del suceso que puede recuperar escribiendo un programa de aplicación MQI adecuado que lleve a cabo los siguientes pasos:

- Obtener el mensaje de la cola.
- Procesar el mensaje para extraer los datos del suceso.

La información relacionada describe el formato de los mensajes de suceso.

Condiciones que provocan sucesos

La siguiente lista ofrece ejemplos de las condiciones que pueden provocar sucesos de instrumentación:

- Se alcanza un límite de umbral para el número de mensajes de la cola.
- Se inicia o detiene una instancia de canal.
- Un gestor de colas pasa a estar activo o se solicita que se detenga.
- Una aplicación intenta abrir una cola especificando un ID de usuario que no está autorizado en sistemas IBM MQ for IBM i, AIX, Linux®, and Windows.
- Se crean, eliminan, modifican o actualizan objetos.
- Un mandato MQSC o PCF se ejecuta satisfactoriamente.
- Un gestor de colas empieza a escribir en una nueva extensión de registro.
- Colocación de un mensaje en la cola de mensajes no entregados, si se cumplen las condiciones del suceso.

Conceptos relacionados

[“Sucesos de rendimiento” en la página 24](#)

Los sucesos de rendimiento están relacionados con las condiciones que pueden afectar al rendimiento de las aplicaciones que utilizan una cola especificada. El ámbito de los sucesos de rendimiento es la cola. Las llamadas **MQPUT** y **MQGET** en una cola no afectan a la generación de sucesos de rendimiento en otra cola.

[“Programa de ejemplo para supervisar sucesos de instrumentación en Multiplatforms” en la página 57](#) **amqsevt** formatea los sucesos de instrumentación que un gestor de colas puede crear y se proporciona con IBM MQ for Multiplatforms. El programa lee mensajes de las colas de sucesos y les da formato de series legibles.

Tipos de suceso

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

Los sucesos de instrumentación de IBM MQ tienen los tipos siguientes:

- Sucesos de gestor de colas
- Sucesos de canal y de puente
- Sucesos de rendimiento
- Sucesos de configuración
- Sucesos de mandatos
- Sucesos del registrador de anotaciones
- Sucesos locales

Para cada gestor de colas, cada categoría de suceso tiene su cola de sucesos propia. Todos los sucesos de dicha categoría generan, como resultado, un mensaje de suceso que se transfiere a la misma cola.

Esta cola de sucesos:

SYSTEM.ADMIN.QMGR.EVENT
SYSTEM.ADMIN.CHANNEL.EVENT
SYSTEM.ADMIN.PERFM.EVENT
SYSTEM.ADMIN.CONFIG.EVENT
SYSTEM.ADMIN.COMMAND.EVENT
SYSTEM.ADMIN.LOGGER.EVENT
SYSTEM.ADMIN.PUBSUB.EVENT

Contiene mensajes de:

Sucesos de gestor de colas
Sucesos de canal
Sucesos de rendimiento
Sucesos de configuración
Sucesos de mandatos
Sucesos del registrador de anotaciones
Obtiene sucesos relacionados con la publicación/suscripción. Sólo se utiliza con multidifusión. Para obtener más información, consulte [Supervisión de aplicaciones de multidifusión](#).

Al incorporar los sucesos de instrumentación en su propia aplicación de gestión del sistema, podrá supervisar las actividades a través de muchos gestores de colas, en muchos nodos diferentes, y para múltiples aplicaciones de IBM MQ. En concreto, puede supervisar todos los nodos del sistema desde un único nodo (para los nodos que den soporte a los sucesos de IBM MQ), tal como se muestra en la [Figura 2](#) en la página 13.

Se puede informar sobre los sucesos de instrumentación a través de un mecanismo de notificación escrito por el usuario a una aplicación de administración, que puede presentar los sucesos a un operador.

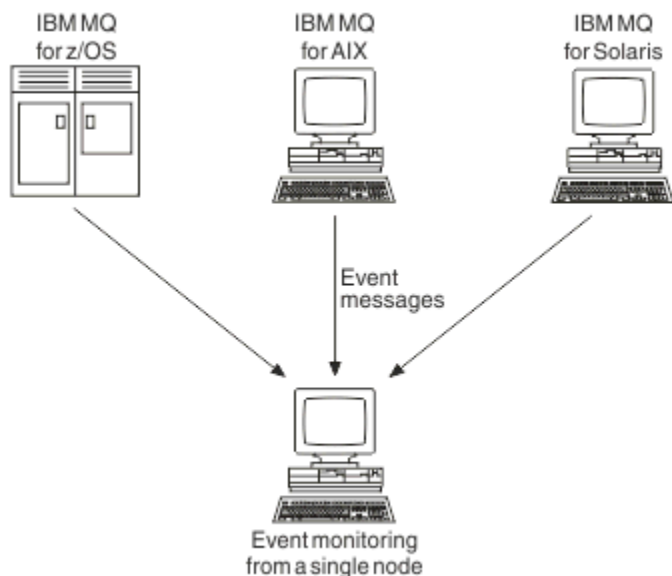


Figura 2. Supervisión de gestores de colas entre plataformas diferentes, en un único nodo

Los sucesos de instrumentación también permiten que las aplicaciones actúen como agentes de otras redes de administración, por ejemplo Tivoli NetView for z/OS, para supervisar informes y crear las alertas pertinentes.

Sucesos de gestor de colas

Los sucesos de gestor de colas están relacionados con el uso de los recursos dentro de los gestores de colas. Por ejemplo, se genera un suceso de gestor de colas si una aplicación intenta transferir un mensaje a una cola que no existe.

Los ejemplos siguientes muestran condiciones que pueden provocar un suceso de gestor de colas:

- Una aplicación emite una llamada MQI que falla. El código de razón de la llamada es el mismo que el código de razón del mensaje de suceso.

Una condición similar puede producirse durante una operación interna de un gestor de colas; por ejemplo, al generar un mensaje de informe. El código de razón de un mensaje de suceso puede coincidir con un código de razón MQI, aunque no este asociado a ninguna aplicación. No presuponga que, ya que un código de razón de mensaje de suceso se parece a un código de razón MQI, el suceso se haya generado necesariamente debido a una llamada MQI fallida, procedente de una aplicación.

- Se emite un mandato a un gestor de colas, y el proceso de este mandato provoca un suceso. Por ejemplo:
 - Se detiene o se inicia gestor de colas.
 - Se emite un mandato en el que el ID de usuario asociado no está autorizado para dicho mandato.

IBM MQ coloca mensajes para sucesos de gestor de colas en el SYSTEM.ADMIN.QMGR.EVENT y da soporte a los siguientes tipos de sucesos del gestor de colas:

ALW Autorización (solo en AIX, Linux, and Windows)

Los sucesos de autorización informan de una autorización como, por ejemplo, una aplicación que intentar abrir una cola para el cual no tiene la autorización necesaria, o un mandato que se emite desde un ID de usuario que no tiene la autorización pertinente. El mensaje de suceso de autorización puede contener los datos de suceso siguientes:

- [No autorizado \(tipo 1\)](#)
- [No autorizado \(tipo 2\)](#)
- [No autorizado \(tipo 3\)](#)
- [No autorizado \(tipo 4\)](#)
- [No autorizado \(tipo 5\)](#)
- [No autorizado \(tipo 6\)](#)

Todos los sucesos de autorización solo son válidos en AIX, Linux, and Windows.

Inhibición

Los sucesos de inhibición indican que una operación MQPUT o MQGET se ha intentado en una cola donde la cola se ha inhibido para las operaciones put (transferir) o get (obtener), o en un tema donde el tema está inhibido para las publicaciones. El mensaje de suceso de inhibición puede contener los datos de suceso siguientes:

- [Obtener inhibido](#)
- [Transferir inhibido](#)

Local

Cuando una aplicación o el gestor de colas no ha sido capaz de acceder a una cola local o un objeto local, por ejemplo, porque el objeto no se ha definido, el gestor de colas puede generar un mensaje de suceso local. El mensaje de suceso local puede contener los datos de suceso siguientes:

- [Error de tipo de cola base de alias](#)
- [Cola base de alias desconocida](#)
- [Nombre de objeto desconocido](#)

Remoto

Cuando una aplicación o el gestor de colas no puede acceder a una cola remota en otro gestor de colas, por ejemplo, la cola de transmisión no se ha definido correctamente, el gestor de colas puede generar un mensaje de suceso remoto. El mensaje de suceso remoto puede contener los datos de sucesos siguientes:

- [Error de tipo de cola de transmisión predeterminada](#)
- [Error de uso de cola de transmisión predeterminada](#)
- [Error de tipo de cola](#)

- [Error de nombre de cola remota](#)
- [Error de tipo de cola de transmisión](#)
- [Error de uso de cola de transmisión](#)
- [Cola de transmisión predeterminada desconocida](#)
- [Gestor de colas remoto desconocido](#)
- [Cola de transmisión desconocida](#)

Iniciar y detener

Los sucesos de inicio y detención indican que se ha iniciado un gestor de colas, o se le ha solicitado que se detenga o se desactive temporalmente.

 z/OS sólo da soporte a los sucesos de inicio.

Los sucesos de detención no se registran, a menos que la persistencia predeterminada de mensajes de la cola SYSTEM.ADMIN.QMGR.EVENT se haya definido como persistente. El mensaje de suceso de inicio y detención puede contener los datos de suceso siguientes:

- [Gestor de colas activo](#)
- [Gestor de colas no activo](#)

Para cada tipo de suceso de esta lista, puede establecer un atributo de gestor de colas para habilitar, o inhabilitar, el tipo de suceso.

Sucesos de canal y de puente

Los canales informan de estos sucesos como resultado de las condiciones detectadas durante su funcionamiento. Por ejemplo, cuando se detiene una instancia de canal.

Los sucesos de canal se generan en las circunstancias siguientes:

- Cuando un mandato inicia o detiene un canal.
- Cuando una se inicia o se detiene instancia de canal.
- Cuando un canal recibe un aviso error de conversión al obtener un mensaje.
- Cuando se intenta crear un canal automáticamente; el suceso se genera si el intento resulta satisfactorio o falla.

Nota: Las conexiones de cliente no provocan los sucesos Canal detenido o Canal iniciado.

Cuando un mandato se utiliza para iniciar un canal, se genera un suceso. Otro suceso se genera cuando se inicia la instancia de canal. No obstante, cuando se inicia un canal mediante un escucha, el mandato **runmqchl**, o un mensaje desencadenante del gestor de colas, no se genera un suceso. En estos casos, se genera un suceso sólo cuando se inicia la instancia de canal.

Un mandato de detención o inicio de canal satisfactorio genera, como mínimo, dos sucesos. Dichos sucesos se generan para los dos gestores de colas que el canal conecta (siempre que den soporte a los sucesos).

Si un suceso de canal se transfiere a una cola de sucesos, una condición de error provoca que el gestor de colas cree un suceso.

Los mensajes de suceso para los sucesos de canal y puente se transfieren a la cola SYSTEM.ADMIN.CHANNEL.EVENT.

Los mensajes de suceso de canal pueden contener los datos de suceso siguientes:

- [Canal activado](#)
- [Error de definición automática de canal](#)
- [Definición automática de canal correcta](#)
- [Error de conversión de canal](#)
- [Canal no activado.](#)
- [Canal iniciado](#)

- [Canal detenido](#)
- [Canal detenido por el usuario](#)
- [Canal bloqueado](#)

Sucesos de puente IMS (sólo z/OS)



Se informa de estos sucesos cuando se inicia o se detiene un puente IMS.

Los mensajes de suceso de puente IMS pueden contener los siguientes datos de suceso:

- [Puente iniciado](#)
- [Puente detenido](#)

Sucesos SSL

El único suceso de TLS es el suceso Error SSL de canal. Se informa de este suceso cuando un canal que utiliza TLS no ha podido establecer una conexión TLS.

Los mensajes de suceso SSL pueden contener los datos de suceso siguientes:

- [Error SSL de canal](#)
- [Aviso SSL de canal](#)

Sucesos de rendimiento

Los sucesos de rendimiento son notificaciones de que un recurso ha alcanzado una condición de umbral. Por ejemplo, se ha alcanzado un límite de profundidad de cola.

Los sucesos de rendimiento están relacionados con las condiciones que pueden afectar al rendimiento de las aplicaciones que utilizan una cola especificada. No se generan para las colas de sucesos en sí.

El tipo de suceso se devuelve en el campo de identificador de mandato de los datos de mensaje.

Si un gestor de colas intenta transferir el mensaje de suceso en un suceso de gestor de colas o mensaje de suceso de rendimiento en una cola de sucesos, y se detecta un error que normalmente crearía un suceso, no se crea otro suceso ni se realiza ninguna acción.

Las llamadas MQGET y MQPUT que aparezcan dentro de una unidad de trabajo pueden generar sucesos de rendimiento, independientemente de si se confirma o se restituye la unidad de trabajo.

Los mensajes de suceso de los sucesos de rendimiento se transfieren a la cola SYSTEM.ADMIN.PERFM.EVENT.

Hay dos tipos de sucesos de rendimiento:

Sucesos de profundidad de cola

Los sucesos de profundidad de cola están relacionados con el número de mensajes de una cola; es decir, hasta qué punto la cola está llena o vacía. Se da soporte a estos sucesos para las colas compartidas. Los mensajes de suceso de profundidad de cola puede contener los datos de suceso siguientes:

- [Profundidad de cola alta](#)
- [Profundidad cola baja](#)
- [Cola llena](#)

Sucesos de intervalo de servicio de cola

Los sucesos de intervalo de servicio de cola se refieren a si los mensajes se procesan dentro de un intervalo de tiempo especificado por el usuario. No se da soporte a estos sucesos para las colas compartidas.



IBM MQ for z/OS da soporte a los sucesos de profundidad de cola para colas QSGDISP (SHARED), pero no a los sucesos de intervalo de servicio. Los sucesos de gestor de colas y de canal

no se ven afectados por las colas compartidas. Los mensajes de suceso de servicio de cola pueden contener los siguientes datos de suceso:

- [Intervalo de servicio de cola alto](#)
- [Intervalo servicio cola correcto](#)

Sucesos de configuración

Los sucesos de configuración se generan cuando se solicita explícitamente un suceso de configuración, o se generan automáticamente al crear, modificar o suprimir un objeto.

Un mensaje de suceso de configuración contiene información sobre los atributos de un objeto. Por ejemplo, se genera un mensaje de suceso de configuración si se crea un objeto de lista de nombres, y contiene información sobre los atributos del objeto de la lista de nombres.

Los mensajes de suceso de los sucesos de configuración se transfieren a la cola SYSTEM.ADMIN.CONFIG.EVENT.

Los sucesos de configuración trabajan de la forma siguiente:

- Un seguimiento de auditoría de los cambios en la configuración del gestor de colas se conserva a través de mensajes de suceso, que se escriben en la cola SYSTEM.ADMIN.CONFIG.EVENT. Puede habilitar estos sucesos para que se escriban a través del parámetro **CONFIGEV** en el mandato [ALTER QMGR](#).
- Estos sucesos se generan cuando un mandato DEFINE, ALTER o DELETE actúa frente a un objeto, o se utiliza una llamada MQSET.
- Puede crear una imagen de línea base de la configuración del gestor de colas actual utilizando el mandato [REFRESH QMGR TYPE\(CONFIGEV\)](#), que creará un mensaje de suceso para cada objeto del gestor de colas. Tenga en cuenta que puesto que esto podría ser una operación que consume tiempo, si tiene muchos objetos, puede dividir la tarea en conjuntos de objetos menores utilizando los calificadores NAME y OBJECT en el mandato.
- El mensaje de suceso registra una de las cuatro posibles razones:
 - MQRC_CONFIG_CHANGE_OBJECT
 - MQRC_CONFIG_CREATE_OBJECT
 - MQRC_CONFIG_DELETE_OBJECT
 - MQRC_CONFIG_REFRESH_OBJECT

MQRC_CONFIG_CHANGE_OBJECT, MQRC_CONFIG_CREATE_OBJECT o MQRC_CONFIG_DELETE_OBJECT se producen para los mandatos MQSC o PCF respectivos que podría emitir frente a un objeto.

MQRC_CONFIG_REFRESH_OBJECT se produce para estos mensajes de suceso escritos, al crear la imagen de línea base.

Existen cuatro tipos de suceso de configuración:

Sucesos de creación de objeto

Los sucesos de creación de objeto se generan cuando se crea un objeto. El mensaje de suceso contiene los siguientes datos de suceso: [Crear objeto](#).

Sucesos de cambio de objeto

Los sucesos de cambio de objeto se generan cuando se modifica un objeto. El mensaje de suceso contiene los siguientes datos de suceso: [Cambiar objeto](#).

Sucesos de supresión de objeto

Los sucesos de supresión de objeto se generan cuando se suprime un objeto. El mensaje de suceso contiene los siguientes datos de suceso: [Suprimir objeto](#).

Sucesos de renovación de objeto

Los sucesos de renovación de objeto los genera una solicitud explícita de renovación. El mensaje de suceso contiene los siguientes datos de suceso: [Renovar objeto](#).

Sucesos de mandatos

Los sucesos de mandato se notifican cuando un mandato MQSC o PCF se ejecuta correctamente.

Un mensaje de suceso de mandato contiene información sobre el origen, el contexto y el contenido de un mandato. Por ejemplo, si el mandato MQSC, ALTER QLOCAL, se ejecuta de forma satisfactoria, se genera un mensaje de suceso de mandato con dicha información.

Los mensajes de suceso de los sucesos de mandato se transfieren a la cola SYSTEM.ADMIN.COMMAND.EVENT.

Los sucesos de mandato contienen los datos de suceso siguientes: [Mandato](#).

Multi Sucesos del registrador de anotaciones

Se informa de los sucesos de registrador cuando un gestor de colas que utiliza registro lineal empieza a grabar registros de anotaciones en una nueva extensión de registro **IBM i** o, en IBM i, en un nuevo destinatario de diario. **z/OS** Los sucesos de registrador no están disponibles con IBM MQ for z/OS.

Un mensaje de suceso de registrador contiene información que especifica las extensiones de registro que necesita el gestor de colas para poder reiniciar el gestor de colas, o para poder llevar a cabo la recuperación desde soporte.

Los mensajes de suceso para los sucesos de registrador se transfieren a la cola SYSTEM.ADMIN.LOGGER.EVENT.

El mensaje de suceso del anotador contiene los siguientes datos de suceso: [Anotador](#).

Resumen de datos de mensaje de suceso

Utilice este resumen para obtener información sobre los datos de suceso que cada tipo de mensaje de suceso puede contener.

Tipo de suceso	Consulte estos temas
Sucesos de autoridad	No autorizado (tipo 1)
	No autorizado (tipo 2)
	No autorizado (tipo 3)
	No autorizado (tipo 4)
	No autorizado (tipo 5)
	No autorizado (tipo 6)
Sucesos de canal	Canal activado
	Error de definición automática de canal
	Definición automática de canal correcta
	Canal bloqueado
	Error de conversión de canal
	Canal no activado.
	Canal iniciado
	Canal detenido
	Canal detenido por el usuario
Sucesos de mandatos	Mandato

Tipo de suceso	Consulte estos temas
Sucesos de configuración	Crear objeto
	Cambiar objeto
	Suprimir objeto
	Renovar objeto
Sucesos de puente IMS	Puente iniciado
	Puente detenido
Inhibir sucesos	Obtener inhibido
	Transferir inhibido
Sucesos locales	Error de tipo de cola base de alias
	Cola base de alias desconocida
	Nombre de objeto desconocido
Sucesos del registrador de anotaciones	Registrador de anotaciones
Sucesos de rendimiento	Profundidad de cola alta
	Profundidad cola baja
	Cola llena
	Intervalo de servicio de cola alto
	Intervalo servicio cola correcto
Sucesos remotos	Error de tipo de cola de transmisión predeterminada
	Error de uso de cola de transmisión predeterminada
	Error de tipo de cola
	Error de nombre de cola remota
	Error de tipo de cola de transmisión
	Error de uso de cola de transmisión
	Cola de transmisión predeterminada desconocida
	Gestor de colas remoto desconocido
	Cola de transmisión desconocida
Sucesos SSL	Error SSL de canal
Iniciar y detener sucesos	Gestor de colas activo
	Gestor de colas no activo

Control de sucesos


Puede habilitar e inhabilitar los sucesos, especificando los valores adecuados para el gestor de colas, atributos de cola, o ambos, dependiendo del tipo de suceso.

Debe habilitar cada suceso de instrumentación que desea que se genere. Por ejemplo, las condiciones que provoca un suceso Cola llena son:

- Los sucesos Cola llena se habilitan para una cola especificada, y

- Una aplicación emite una solicitud MQPUT para transferir un mensaje a dicha cola, pero la solicitud falla porque la cola está llena.

Habilite e inhabilite los sucesos mediante una de las técnicas siguientes:

- Mandatos de script de IBM MQ (MQSC).
- Los mandatos PCF correspondientes de IBM MQ.
-  Las operaciones y los paneles de control de los gestores de colas en z/OS.
- IBM MQ Explorer.

Nota: Puede establecer atributos relacionados con los sucesos tanto de las colas y los gestores de colas, sólo mediante un mandato. La llamada MQI MQSET no da soporte a los atributos relacionados con los sucesos.

Conceptos relacionados

“Sucesos de instrumentación” en la página 10

Un suceso de instrumentación es una combinación lógica de condiciones que un gestor de colas o una instancia de canal detecta y transfiere un mensaje especial, denominado *mensaje de suceso*, a una cola de sucesos.

Uso de los paneles de control y operaciones en z/OS

Tareas relacionadas

Automatización de tareas de administración

Utilización de formatos de mandato programables

Referencia relacionada

“Tipos de suceso” en la página 12

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

Mandatos MQSC

Control de sucesos del gestor de colas

Puede controlar los sucesos de gestor de colas utilizando atributos de gestor de colas. Para habilitar los sucesos de gestor de colas, establezca el atributo de gestor de colas adecuado en ENABLED. Para inhabilitar los sucesos de gestor de colas, establezca el atributo de gestor de colas adecuado en DISABLED.

Para habilitar o inhabilitar sucesos de gestor de colas, utilice el mandato MQSC **ALTER QMGR**, especificando el atributo de gestor de colas apropiado. En la Tabla 1 en la página 20 se resume cómo habilitar sucesos de gestor de colas. Para inhabilitar un suceso de gestor de colas, establezca el parámetro pertinente en DISABLED.


<i>Tabla 1. Habilitación de sucesos de gestor de colas mediante mandatos MQSC</i>	
Suceso	Parámetro ALTER QMGR
Autorización	AUTHOREV (ENABLED)
Inhibición	INHIBTEV (ENABLED)
Local	LOCALEV (ENABLED)
Remoto	REMOTEEV (ENABLED)
Iniciar y detener	STRSTPEV (ENABLED)

Control de sucesos de canal y puente

Puede controlar los sucesos de canal mediante los atributos de gestor de colas. Para habilitar los sucesos de canal, establezca el atributo adecuado del gestor de colas en ENABLED. Para inhabilitar los sucesos de canal, establezca el atributo adecuado del gestor de colas en DISABLED.

Para habilitar o inhabilitar los sucesos de canal, utilice el mandato MQSC **ALTER QMGR**, especificando el atributo de gestor de colas apropiado. En la Tabla 2 en la página 21 se resume cómo habilitar sucesos

de canal y de puente. Para inhabilitar un suceso de gestor de colas, establezca el parámetro pertinente en DISABLED.

Restricción:  Los sucesos de definición automática de canal no están disponibles en IBM MQ for z/OS.

<i>Tabla 2. Habilitación de sucesos de canal y de puente mediante mandatos MQSC</i>	
Suceso	Parámetro ALTER QMGR
Canal Relacionado sólo con errores de canal Puente IMS SSL Definición automática de canal	CHLEV (ENABLED) CHLEV (EXCEPTION) BRIDGEEV (ENABLED) SSLEV (ENABLED) CHADEV(ENABLED)

Con CHLEV establecido en excepción, se generan los siguientes códigos de retorno y los calificadores de razón correspondientes:

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_ERROR
- MQRC_CHANNEL_NOT_ACTIVATED
- MQRC_CHANNEL_STOPPED
 - con los siguientes calificadores de razón:
 - MQRQ_CHANNEL_STOPPED_ERROR
 - MQRQ_CHANNEL_STOPPED_RETRY
 - MQRQ_CHANNEL_STOPPED_DISABLED
- MQRC_CHANNEL_STOPPED_BY_USER
- MQRC_CHANNEL_BLOCKED
 - con los siguientes calificadores de razón:
 - MQRQ_CHANNEL_BLOCKED_NOACCESS
 - MQRQ_CHANNEL_BLOCKED_USERID
 - MQRQ_CHANNEL_BLOCKED_ADDRESS

Control de sucesos de rendimiento

Puede controlar los sucesos de rendimiento mediante el atributo de gestor de colas PERFMEV. Para habilitar los sucesos de rendimiento, establezca PERFMEV en ENABLED. Para inhabilitar los sucesos de rendimiento, establezca el atributo de gestor de colas adecuado en DISABLED.

Para establecer el atributo de gestor de colas PERFMEV en ENABLED, utilice el mandato MQSC siguiente:

```
ALTER QMGR PERFMEV (ENABLED)
```

Para habilitar sucesos de rendimiento específicos, establezca el atributo de cola adecuado. Además, especifique las condiciones que provoquen el suceso.

Sucesos de profundidad de cola

De forma predeterminada, están inhabilitados todos los sucesos de profundidad de cola. Para configurar una cola para cualquiera de los sucesos de profundidad de cola:

1. Habilite los sucesos de rendimiento en el gestor de colas.
2. Habilite el suceso en la cola pertinente.
3. Establezca los límites, si es necesario, en los niveles apropiados, expresados como porcentaje de la profundidad de cola máxima.

Sucesos de intervalo de servicio de cola

Para configurar una cola para los sucesos de intervalo de servicio de cola, debe:

1. Habilite los sucesos de rendimiento en el gestor de colas.
2. Establecer el atributo de control para un suceso intervalo de servicio de cola alto o correcto en la cola, según convenga.
3. Especificar el intervalo de tiempo de servicio estableciendo el atributo QSVCINT para la cola, en la longitud de tiempo pertinente.

Nota: Cuando está habilitado, un suceso de intervalo de servicio de cola se puede generar en cualquier momento que resulte adecuado, no hace falta esperar necesariamente a que se emita una llamada MQI para la cola. No obstante, si se utiliza una llamada MQI en una cola para transferir o eliminar un mensaje, en ese momento se generará cualquier suceso de rendimiento que resulte aplicable. El suceso no se genera cuando el tiempo transcurrido pasa a ser igual al intervalo de servicio.

Control de sucesos de configuración, mandato y registrador

Puede controlar los sucesos de configuración, mandato y registrador mediante los atributos de gestor de colas CONFIGEV, CMDEV y LOGGEREV. Para habilitar estos sucesos, establezca el atributo de gestor de colas adecuado en ENABLED. Para inhabilitar estos sucesos, establezca el atributo adecuado del gestor de colas en DISABLED.

Sucesos de configuración

Para habilitar los sucesos de configuración, establezca CONFIGEV en ENABLED. Para inhabilitar los sucesos de configuración, establezca CONFIGEV en DISABLED. Por ejemplo, puede habilitar los sucesos de configuración mediante el mandato MQSC siguiente:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Sucesos de mandatos

Para habilitar los sucesos de mandato, establezca CMDEV en ENABLED. Para habilitar los sucesos de mandato para todos los mandatos excepto los mandatos MQSC DISPLAY y los mandatos PCF Inquire, establezca CMDEV en NODISPLAY. Para inhabilitar los sucesos de mandato, establezca CMDEV en DISABLED. Por ejemplo, puede habilitar los sucesos de mandato mediante el mandato MQSC siguiente:

```
ALTER QMGR CMDEV (ENABLED)
```

Sucesos del registrador de anotaciones

Para habilitar los sucesos de registrador, establezca LOGGEREV en ENABLED. Para inhabilitar los sucesos de registrador, establezca LOGGEREV en DISABLED. Por ejemplo, puede habilitar los sucesos de registrador mediante el mandato MQSC siguiente:

```
ALTER QMGR LOGGEREV(ENABLED)
```

Colas de sucesos

Cuando se produce un suceso, el gestor de colas coloca un mensaje de suceso en la cola de sucesos definida. El mensaje de suceso contiene información sobre el suceso.

Puede definir colas de sucesos como:

- Colas locales
- Colas alias
- Definiciones locales de colas remotas o como
- Colas de clúster remoto

Si define todas las colas de sucesos como definiciones locales de la misma cola remota de un gestor de colas, puede centralizar las actividades de supervisión.

No debe definir colas de sucesos como colas de transmisión, ya que los mensajes de suceso tienen formatos que son incompatibles con el formato de mensaje que se necesita para las colas de transmisión.

Las colas de sucesos compartidas son colas locales definidas con el valor QSGDISP(SHARED).

Para obtener más información sobre la definición de colas compartidas en z/OS, consulte [Programación de aplicaciones con colas compartidas](#).

Cuándo una cola de sucesos no está disponible

Si se produce un suceso cuando la cola de sucesos no está disponible, el mensaje de suceso se pierde. Por ejemplo, si no define una cola de sucesos para una categoría de suceso, se pierden todos los mensajes de suceso para dicha categoría. Los mensajes de suceso no se guardan, por ejemplo, en la cola de mensajes no entregados.

No obstante, puede definir la cola de sucesos como una cola remota. A continuación, si surge un problema en el sistema remoto que transfiere los mensajes a la cola resuelta, el mensaje de suceso llega a la cola de mensajes no entregados del sistema remoto.

Una cola de sucesos puede no estar disponible por muchas razones diferentes, entre las que se incluyen las siguientes:

- La cola no se ha definido.
- La cola se ha suprimido.
- La cola está llena.
- La cola se ha inhibido para las operaciones de tipo put.

La ausencia de una cola de sucesos no impide que se produzca el suceso. Por ejemplo, después de un suceso de rendimiento, el gestor de colas cambia los atributos de la cola, y restablece las estadísticas de la misma. Este cambio se produce independientemente de si el mensaje de suceso se coloca o no en la cola de sucesos de rendimiento. Lo mismo ocurre en el caso de los sucesos de configuración y de mandato.

Utilización de las colas de sucesos desencadenantes

Puede configurar las colas de sucesos con desencadenantes, de forma que cuando se genere un suceso, el mensaje de suceso que se transfiera a la cola de sucesos inicie una aplicación de supervisión escrita por el usuario. Esta aplicación puede procesar los mensajes de suceso y llevar a cabo la acción adecuada. Por ejemplo, es posible que ciertos acontecimientos necesiten informar a un operador, otros sucesos pueden iniciar una aplicación que realice algunas tareas de administración automáticamente.

Las colas de sucesos pueden tener asociadas acciones de desencadenante, y pueden crear mensajes de desencadenante. No obstante, si estos mensajes de desencadenante, a su vez, provocan condiciones que normalmente generarían un suceso, no se genera ningún suceso. Al no generarse ningún suceso en esta instancia, se garantiza que no se produzca el bucle.

Conceptos relacionados

[“Control de sucesos” en la página 19](#)

Puede habilitar e inhabilitar los sucesos, especificando los valores adecuados para el gestor de colas, atributos de cola, o ambos, dependiendo del tipo de suceso.

[“Formato de mensajes de sucesos” en la página 24](#)

Los mensajes de suceso contienen información sobre un suceso y su causa. Al igual que otros mensajes de IBM MQ, un mensaje de suceso consta de dos partes: un descriptor de mensaje y los datos de mensaje.

[Programación de aplicaciones con colas compartidas](#)

[Condiciones para un suceso desencadenante](#)

Referencia relacionada

[QSGDisp \(MQLONG\)](#)

Formato de mensajes de sucesos

Los mensajes de suceso contienen información sobre un suceso y su causa. Al igual que otros mensajes de IBM MQ, un mensaje de suceso consta de dos partes: un descriptor de mensaje y los datos de mensaje.

- El descriptor de mensaje se basa en la estructura MQMD.
- Los datos del mensaje constan de una *cabecera de suceso* y los *datos de suceso*. La cabecera de suceso contiene el código de razón que identifica el tipo de suceso. Transferir el mensaje de suceso y cualquier acción posterior que se lleve a cabo, no afecta al código de razón que devuelve la llamada MQI que haya provocado el suceso. Los datos de suceso proporcionan información adicional sobre el suceso.

Normalmente, los mensajes de suceso se procesan con una aplicación de gestión de sistemas personalizada, para que se cumplan los requisitos de la empresa en la que se ejecute.

Cuando los gestores de colas de un grupo de compartición de colas detectan las condiciones para generar un mensaje de suceso, varios gestores de colas pueden generar un mensaje de suceso para la cola compartida, dando como resultado varios mensajes de suceso. Para asegurar que un sistema pueda correlacionar varios mensajes de suceso de gestores de colas diferentes, dichos mensajes de suceso tienen un identificador de correlación exclusivo (*CorrelId*) que se establece en el descriptor de mensaje (MQMD).

Referencia relacionada

[“MQMD de informe de actividad \(descriptor de mensaje\)” en la página 103](#)

Utilice esta página para ver los valores contenidos en la estructura MQMD de un informe de actividad.

[“MQEPH de informe de actividad \(cabecera PCF incorporada\)” en la página 107](#)

Utilice esta página para ver los valores contenidos en la estructura MQEPH de un informe de actividad.

[“MQCFH de informe de actividad \(cabecera PCF\)” en la página 108](#)

Utilice esta página para ver los valores PCF contenidos en la estructura MQCFH de un informe de actividad.

[Referencia de mensajes de suceso](#)

[Formato de mensaje de suceso](#)

[Un MQMD \(descriptor de mensaje\) de mensaje de suceso](#)

[MQCFH \(cabecera PCF\) de mensaje de suceso](#)

[Descripciones de mensajes de suceso](#)

Sucesos de rendimiento

Los sucesos de rendimiento están relacionados con las condiciones que pueden afectar al rendimiento de las aplicaciones que utilizan una cola especificada. El ámbito de los sucesos de rendimiento es la cola. Las llamadas **MQPUT** y **MQGET** en una cola no afectan a la generación de sucesos de rendimiento en otra cola.

Los mensajes de suceso de rendimiento se pueden generar en cualquier momento que resulte adecuado, no hace falta esperar necesariamente a que se emita una llamada MQI para la cola. No obstante, si utiliza una llamada MQI en una cola para transferir o eliminar un mensaje, en ese momento se generará cualquier suceso de rendimiento que resulte aplicable.

Todos los mensajes de suceso de rendimiento que se generan se colocan en la cola SYSTEM.ADMIN.PERFM.EVENT.

Los datos de suceso contienen un código de razón que identifica la causa del suceso, un conjunto de estadísticas de sucesos de rendimiento y otros datos. En la lista siguiente se describen los tipos de datos de suceso que se pueden devolver en los mensajes de suceso de rendimiento:

- [Profundidad de cola alta](#)
- [Profundidad cola baja](#)
- [Cola llena](#)
- [Intervalo de servicio de cola alto](#)

- [Intervalo servicio cola correcto](#)

En los ejemplos que ilustran el uso de sucesos de rendimiento se presupone que ha establecido los atributos de colas mediante los mandatos adecuados de IBM MQ (MQSC). En z/OS, también puede establecer los atributos de cola mediante las operaciones y los paneles de controles de los gestores de colas.

Referencia relacionada

[“Tipos de suceso” en la página 12](#)

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.


Estadísticas de sucesos de rendimiento

Los datos de suceso de rendimiento del mensaje de suceso contienen estadísticas sobre el suceso. Utilice las estadísticas para analizar el comportamiento de una cola especificada.

Los datos de suceso del mensaje de suceso contienen información sobre el suceso para los programas de gestión del sistema. Para todos los sucesos de rendimiento, los datos de suceso contienen los nombres del gestor de colas y de la cola asociados con el suceso. Los datos de suceso también contienen estadísticas relacionadas con el suceso. En la [Tabla 3 en la página 25](#) se resumen las estadísticas de sucesos que puede utilizar para analizar el comportamiento de una cola. Todas las estadísticas se refieren a lo que haya ocurrido desde la última vez que se restablecieron las estadísticas.

<i>Tabla 3. Estadísticas de sucesos de rendimiento</i>	
Parámetro	Descripción
TimeSinceReset	El tiempo transcurrido desde que se restablecieron las estadísticas por última vez.
HighQDepth	El número máximo de mensajes que hay en la cola desde que se restablecieron las estadísticas por última vez.
MsgEnqCount	El número de mensajes colocados en la cola (el número de llamadas MQPUT efectuadas a la cola), desde que se restablecieron las estadísticas por última vez.
MsgDeqCount	El número de mensajes eliminados de la cola (el número de llamadas MQGET efectuadas a la cola), desde que se restablecieron las estadísticas por última vez.

Las estadísticas de suceso de rendimiento se restablecen cuando se efectúa alguno de los cambios siguientes:

- Se produce un suceso de rendimiento (se restablecen las estadísticas en todos los gestores de colas activos).
- Un gestor de colas se detiene y se reinicia.
- Se emite el mandato PCF, Restablecer estadísticas de cola, desde un programa de aplicación.
-  Sólo en z/OS, se emite el mandato RESET QSTATS en la consola.

Conceptos relacionados

[“Sucesos de rendimiento” en la página 24](#)

Los sucesos de rendimiento están relacionados con las condiciones que pueden afectar al rendimiento de las aplicaciones que utilizan una cola especificada. El ámbito de los sucesos de rendimiento es la cola. Las llamadas **MQPUT** y **MQGET** en una cola no afectan a la generación de sucesos de rendimiento en otra cola.

[“El temporizador de servicio” en la página 27](#)

Los sucesos de intervalo de servicio de cola utilizan un temporizador interno, denominado *temporizador de servicio*, que se controla a través del gestor de colas. El temporizador de servicio sólo se utiliza si se habilita un suceso de intervalo de servicio de cola.

[“Reglas de sucesos de intervalo de servicio de cola” en la página 28](#)

Las reglas formales controlan cuándo está establecido el temporizador de servicio, y se generan los sucesos de intervalo de servicio de cola.

Tareas relacionadas

[“Habilitación de sucesos de intervalo de servicio de cola” en la página 29](#)

Para configurar una cola para los sucesos de intervalo de servicio de cola, debe establecer los atributos de gestor de colas y de cola pertinentes.

Referencia relacionada

Profundidad de cola alta

[Restablecer estadísticas de la cola](#)

[RESET QSTATS](#)

Sucesos de intervalo de servicio de cola

Los sucesos de intervalo de servicio de cola indican si se ha realizado una operación en una cola dentro de un intervalo de tiempo definido por el usuario, que se conoce como el *intervalo de servicio*. En función de la instalación, puede utilizar los sucesos de intervalo de servicio de cola para supervisar si los mensajes se toman de las colas con suficiente rapidez.

Los sucesos de intervalo de servicio de cola no están soportados en colas compartidas.

Pueden producirse los tipos de sucesos de intervalo de servicio de cola siguientes, donde el término *operación get* hace referencia a una llamada **MQGET** o a una actividad que elimina un mensaje de una cola como, por ejemplo, utilizar el mandato **CLEAR QLOCAL**:

Intervalo de servicio de cola correcto

Indica que después de una de las operaciones siguientes:

- Una llamada MQPUT
- Una operación get que deja una cola no vacía

se ha realizado una operación get dentro de un periodo de tiempo definido por el usuario, conocido como *intervalo de servicio*.

Sólo una operación get puede provocar el mensaje de suceso de intervalo de servicio de cola correcto. Los sucesos de intervalo de servicio de cola correcto a veces se describen como sucesos Correctos.

Intervalo de servicio de cola alto

Indica que después de una de las operaciones siguientes:

- Una llamada MQPUT
- Una operación get que deja una cola no vacía

se ha realizado una operación get **fuera** de un periodo de tiempo definido por el usuario.

Bien una operación get o una llamada MQPUT puede provocar el mensaje de suceso de intervalo de servicio de cola Alto. Los sucesos de intervalo de servicio de cola Alto a veces se describen como sucesos Altos.

Para habilitar tanto los sucesos de intervalo de servicio de cola Correcto como los de intervalo de servicio de cola Alto, establezca el atributo de control QServiceIntervalEvent en High. Los sucesos de intervalo de servicio de cola correcto se habilitan automáticamente cuando se genera un suceso de intervalo de servicio de cola Alto. No es necesario habilitar los sucesos de intervalo de servicio de cola correcto de forma independiente.

Los sucesos de tipo Correctos y Altos se excluyen mutuamente, de forma que si se ha habilitado uno, el otro se inhabilita. No obstante, ambos sucesos se pueden inhabilitar simultáneamente.

En la [Figura 3 en la página 27](#) se muestra un gráfico de profundidad de cola en relación al tiempo. En el momento P1, una aplicación emite una llamada MQPUT, para transferir un mensaje a la cola. En el momento G1, otra aplicación emite una llamada MQGET para eliminar el mensaje de la cola.

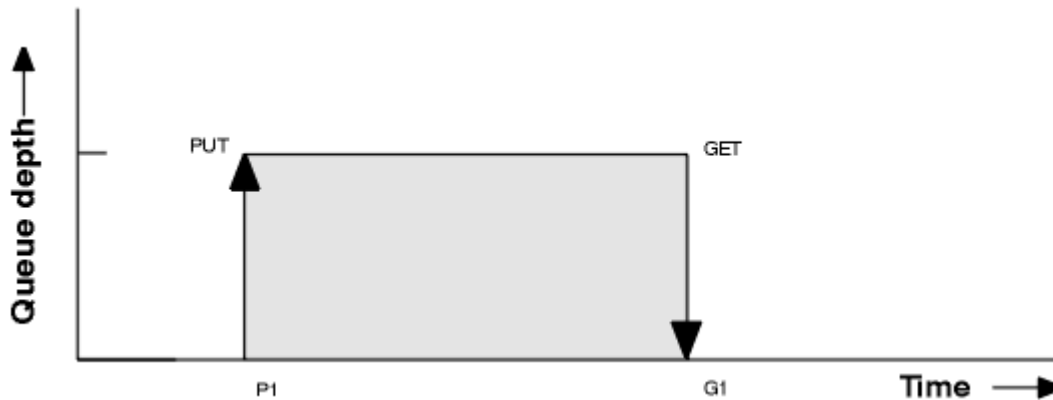


Figura 3. Comprensión de los sucesos de intervalo de servicio de cola

Los resultados posibles de los sucesos de intervalo de servicio de cola son los siguientes:

- Si el tiempo transcurrido entre la transferencia y la obtención es menor o igual que el intervalo de servicio:
 - Se genera un *intervalo de servicio de cola correcto* en el momento G1, si los sucesos de intervalo de servicio de cola están habilitados.
- Si el tiempo transcurrido entre la transferencia y la obtención es mayor o igual que el intervalo de servicio:
 - Se genera un *intervalo de servicio de cola alto* en el momento G1, si los sucesos de intervalo de servicio de cola están habilitados.

El algoritmo para iniciar el temporizador de servicio y generar sucesos, se describe en [“Reglas de sucesos de intervalo de servicio de cola”](#) en la página 28.

Referencia relacionada

[Intervalo de servicio de cola correcto](#)

[Intervalo de servicio de cola alto](#)

[QServiceIntervalEvent \(MQLONG\)](#)

[QServiceIntervalEvent \(entero con signo de 10 dígitos\)](#)

El temporizador de servicio

Los sucesos de intervalo de servicio de cola utilizan un temporizador interno, denominado *temporizador de servicio*, que se controla a través del gestor de colas. El temporizador de servicio sólo se utiliza si se habilita un suceso de intervalo de servicio de cola.

¿Qué hace exactamente la medición del temporizador de servicio?

El temporizador de servicio mide el tiempo transcurrido entre una llamada MQPUT a una cola vacía o una operación de tipo get, y la operación put o get siguiente, siempre que la profundidad de cola no sea cero entre dichas operaciones.

¿Cuándo está activo el temporizador de servicio?

El temporizador de servicio está activa (en ejecución), si la cola contiene mensajes (la profundidad es distinta de cero) y se ha habilitado un suceso de intervalo de servicio de cola. Si la cola se queda vacía (la profundidad es cero), el temporizador se coloca en un estado OFF.

¿Cuándo se restablece el temporizador de servicio?

El temporizador de servicio siempre se restablece después de una operación get. También se restablece mediante una llamada MQPUT a una cola vacía. No obstante, no se restablece necesariamente en un suceso de intervalo de servicio de cola.

¿Cómo se utiliza el temporizador de servicio?

Después de una operación `get` o de una llamada `MQPUT`, el gestor de colas compara el tiempo transcurrido tal como lo ha medido el temporizador de servicio, con el intervalo de servicio definido por el usuario. El resultado de esta comparación es que:

- Se genera un suceso OK si hay una operación `get` y el tiempo transcurrido es menor o igual que el intervalo de servicio, Y este suceso está habilitado.
- Se genera un suceso de tipo Alto si el tiempo transcurrido es mayor que el intervalo de servicio, Y este suceso está habilitado.

Las aplicaciones, ¿pueden leer el temporizador de servicio?

No, el temporizador de servicio es un temporizador interno que no está disponible para las aplicaciones.

¿Qué ocurre con el parámetro *TimeSinceReset* ?

El parámetro *TimeSinceReset* se devuelve como parte de la estadística de sucesos en los datos de suceso. Especifica el tiempo transcurrido entre sucesos de intervalo de servicio de cola sucesivos, a menos que se restablezcan las estadísticas de suceso.

Reglas de sucesos de intervalo de servicio de cola

Las reglas formales controlan cuándo está establecido el temporizador de servicio, y se generan los sucesos de intervalo de servicio de cola.

Reglas del temporizador de servicio

El temporizador de servicio se restablece en cero y se reinicia, tal como se indica a continuación:

- Después de una llamada `MQPUT` a una cola vacía.
- Después de una llamada `MQGET`, si la cola no está vacía después de la llamada `MQGET`.

El restablecimiento del temporizador no depende de si se ha generado un suceso.

Durante el arranque del gestor de colas, el temporizador de servicio se establece en la hora de inicio, si la profundidad de cola es mayor que cero.

Si la cola está vacía después de una operación `get`, el temporizador se coloca en un estado OFF.

Sucesos de intervalo de servicio de cola Alto

El suceso de intervalo de servicio de cola debe estar habilitado (establecido en HIGH).

Los sucesos de intervalo de servicio de cola alto se habilitan automáticamente cuando se genera un suceso de intervalo de servicio de cola correcto.

Si el tiempo de servicio es mayor que el intervalo de servicio, se genera un suceso en, o antes de, la siguiente operación `MQPUT` o de obtención.

Sucesos de tipo correcto de intervalo de servicio de cola

Los sucesos de intervalo de servicio de cola correcto se habilitan automáticamente cuando se genera un suceso de intervalo de servicio de cola Alto.

Si el tiempo de servicio (tiempo transcurrido) es menor o igual que el intervalo de servicio, se genera un suceso en, o antes de, la siguiente operación `get`.

Tareas relacionadas

[“Habilitación de sucesos de intervalo de servicio de cola” en la página 29](#)

Para configurar una cola para los sucesos de intervalo de servicio de cola, debe establecer los atributos de gestor de colas y de cola pertinentes.

Habilitación de sucesos de intervalo de servicio de cola

Para configurar una cola para los sucesos de intervalo de servicio de cola, debe establecer los atributos de gestor de colas y de cola pertinentes.

Acerca de esta tarea

Los sucesos de tipo Alto y Correcto se excluyen mutuamente; es decir, cuando uno está habilitado, el otro se inhabilita automáticamente:

- Cuando en una cola se genera un suceso de tipo alto, el gestor de colas inhabilita los sucesos de tipo Alto, y habilita los sucesos de tipo Correcto para dicha cola automáticamente.
- Cuando en una cola se genera un suceso de tipo Correcto, el gestor de colas inhabilita los sucesos de tipo Correcto, y habilita los sucesos de tipo Alto para dicha cola automáticamente.

Suceso de intervalo de servicio de cola	Atributos de colas
Intervalo de servicio de cola alto Intervalo de servicio de cola correcto Ningún suceso de intervalo de servicio de cola	QSVCI EV (HIGH) QSVCI EV (OK) QSVCI EV(NONE)
Intervalo de servicio	QSVCI NT (<i>tt</i>) donde <i>tt</i> es el tiempo de intervalo de servicio en milisegundos.

Efectúe los pasos siguientes para habilitar los sucesos de intervalo de servicio de cola:

Procedimiento

1. Establezca el atributo de gestor de colas **PERFMEV** en ENABLED.
Los sucesos de rendimiento se habilitan en el gestor de colas.
2. Establezca el atributo de control, **QSVCI EV**, para un suceso correcto o de alta probabilidad de intervalo de servicio de cola en la cola, según convenga.
3. Establezca el atributo **QSVCI NT** para la cola para especificar el tiempo de intervalo de servicio apropiado.

Ejemplo

Para habilitar los sucesos de intervalo de servicio de cola Alto con un tiempo de intervalo de servicio de 10 segundos (10.000 milisegundos) utilice los mandatos MQSC siguientes:

```
ALTER QMGR PERFMEV(ENABLED)  
ALTER QLOCAL('MYQUEUE') QSVCI NT(10000) QSVCI EV(HIGH)
```

Ejemplos de sucesos de intervalo de servicio de cola

Utilice los ejemplos de esta sección para comprender la información que puede obtener de los sucesos de intervalo de servicio de cola.

En los tres ejemplos del subtema se proporcionan ilustraciones, progresivamente más complejas, de la utilización de los sucesos de intervalo de servicio de cola.

Las figuras que acompañan a los ejemplos de cada subtema tienen la misma estructura:

- La figura 1 es un gráfico de profundidad de cola en el tiempo, que muestra llamadas MQGET individuales y llamadas MQPUT.

- En la sección Comentario se muestra una comparación de restricciones de tiempo. Aparecen tres periodos de tiempo que debe tener en cuenta:
 - El intervalo de servicio definido por el usuario.
 - El tiempo que ha medido el temporizador de servicio.
 - El tiempo transcurrido desde la última vez que se restablecieron las estadísticas de suceso (TimeSinceReset de los datos del suceso).
- El resumen Estadísticas de sucesos muestra qué sucesos están habilitados en un momento determinado y qué sucesos se generan.

En los ejemplos se ilustran los aspectos siguientes de los sucesos de intervalo de servicio de cola:

- Cómo la profundidad de cola varía con el tiempo.
- Cómo el tiempo transcurrido según lo mide el temporizador de servicio se compara con el intervalo de servicio.
- Qué suceso está habilitado.
- Qué sucesos se generan.

Recuerde: En el Ejemplo 1 se muestra un caso sencillo en que los mensajes son intermitentes y cada mensaje se elimina de la cola antes de que llegue el siguiente. A partir de los datos de suceso, sabe que el número máximo de mensajes de la cola era uno. Por tanto, puede averiguar cuánto tiempo permaneció cada mensaje en la cola.

No obstante, por lo general, donde haya más de un mensaje en la cola y la secuencia de llamadas MQGET y MQPUT no es predecible, no podrá utilizar los sucesos de intervalo de servicio de cola para calcular cuánto tiempo permanece en una cola un mensaje individual. El parámetro TimeSinceReset, que se devuelve en los datos de suceso, puede incluir una proporción de tiempo cuando no haya mensajes en la cola. Por tanto, para los resultados que se derivan de estas estadísticas se utiliza implícitamente un promedio, para poder incluir estos tiempos.

Conceptos relacionados

[“Sucesos de intervalo de servicio de cola” en la página 26](#)

Los sucesos de intervalo de servicio de cola indican si se ha realizado una operación en una cola dentro de un intervalo de tiempo definido por el usuario, que se conoce como el *intervalo de servicio*. En función de la instalación, puede utilizar los sucesos de intervalo de servicio de cola para supervisar si los mensajes se toman de las colas con suficiente rapidez.

[“El temporizador de servicio” en la página 27](#)

Los sucesos de intervalo de servicio de cola utilizan un temporizador interno, denominado *temporizador de servicio*, que se controla a través del gestor de colas. El temporizador de servicio sólo se utiliza si se habilita un suceso de intervalo de servicio de cola.

Sucesos de intervalo de servicio de cola: ejemplo 1

Una secuencia básica de llamadas MQGET y MQPUT, en las que la profundidad de cola siempre es uno o cero.

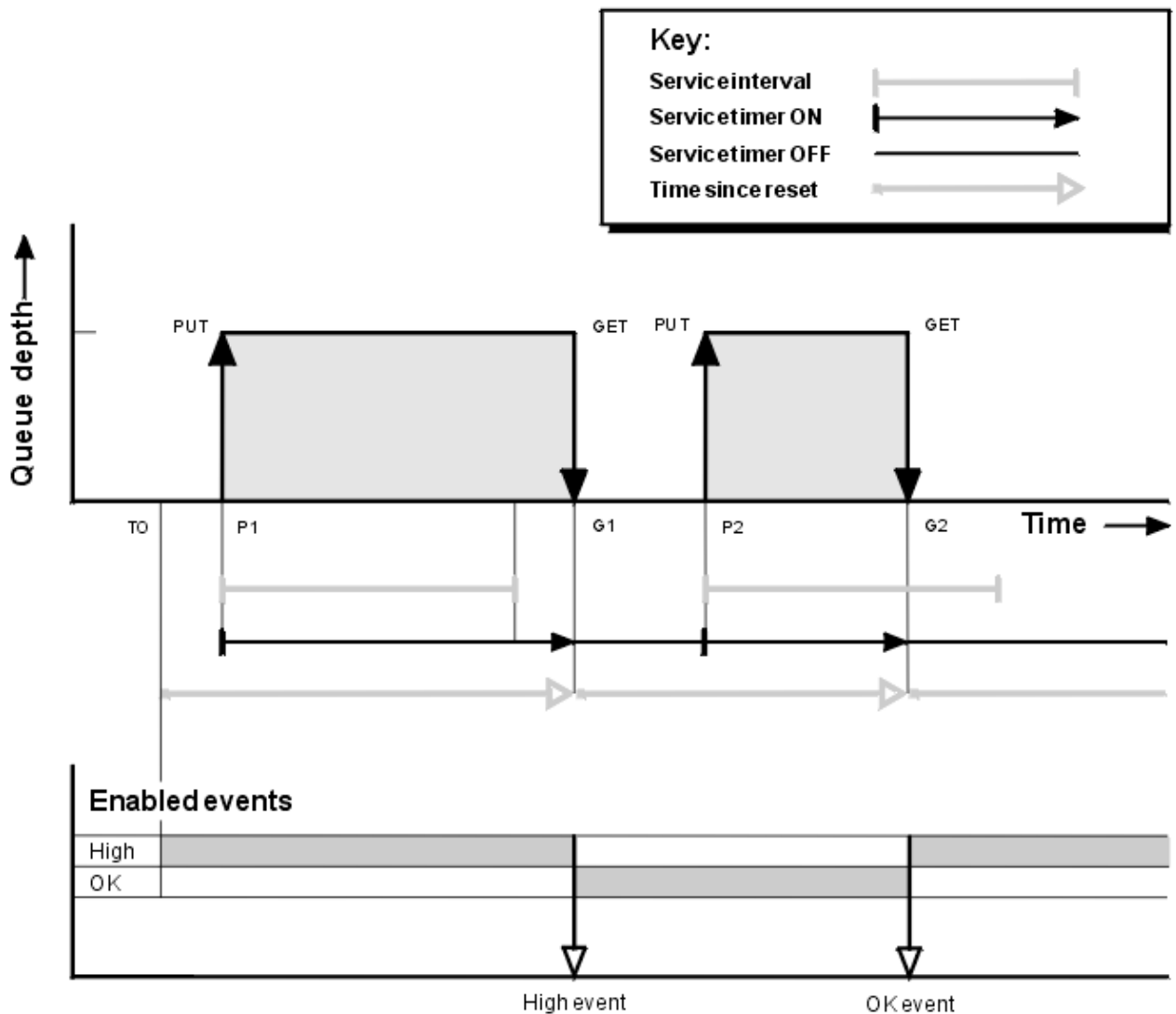


Figura 4. Sucesos de intervalo de servicio de cola: ejemplo 1

Comentario

1. En P1, una aplicación transfiere un mensaje a una cola vacía. Esto inicia el temporizador de servicio. Tenga en cuenta que T0 podría ser la hora de inicio del gestor de colas.
2. En G1, otra aplicación obtiene el mensaje de la cola. Puesto que el tiempo transcurrido entre P1 y G1 es mayor que el intervalo de servicio, se genera un suceso de intervalo de servicio de cola Alto en la llamada MQGET, en G1. Cuando se genera el suceso de tipo Alto, el gestor de colas restablece el atributo de control de sucesos, de forma que:
 - a. El suceso de tipo Correcto se habilita automáticamente.
 - b. El suceso de tipo Alto se inhabilita.
Puesto que la cola ahora está vacía, el temporizador de servicio pasa a un estado OFF.
3. En P2, se transfiere un segundo mensaje a la cola. Esto reinicia el temporizador de servicio.
4. En G2, el mensaje se elimina de la cola. No obstante, puesto que el tiempo transcurrido entre P2 y G2 es menor que el intervalo de servicio, se genera un suceso de intervalo de servicio de cola Correcto en la llamada MQGET, en G2. Cuando se genera el suceso de tipo Correcto, el gestor de colas restablece el atributo de control, de forma que:
 - a. El suceso de tipo Alto se habilita automáticamente.

b. El suceso de tipo Correcto se inhabilita.

Puesto que la cola está vacía, el temporizador de servicio vuelve a pasar a un estado OFF.

Resumen de estadísticas de sucesos

En la [Tabla 5](#) en la [página 32](#) se resumen las estadísticas de sucesos de este ejemplo.

Atributo	Suceso 1	Suceso 2
Hora del suceso	T(G1)	T(G2)
Tipo de suceso	Alto	Correcto
TimeSinceReset	$T(G1) - T(0)$	$T(G2) - T(G1)$
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

La parte central de la [Figura 4](#) en la [página 31](#) se muestra el tiempo transcurrido medido por el temporizador de servicio comparado con el intervalo de servicio de dicha cola. Para ver si se puede producir un suceso de intervalo de servicio de cola, compare la longitud de la línea horizontal que representa el temporizador de servicio (con una flecha) con la de la línea que representa el intervalo de servicio. Si la línea del temporizador de servicio es más larga, y el suceso de intervalo de servicio de cola Alto está habilitado, se genera un suceso de intervalo de servicio de cola Alto en la siguiente obtención. Si la línea del temporizador de servicio es más corta, y el suceso de intervalo de servicio de cola Correcto está habilitado, se genera un suceso de intervalo de servicio de cola Correcto en la siguiente obtención.

Sucesos de intervalo de servicio de cola: ejemplo 2

Una secuencia de llamadas MQPUT y MQGET, en las que la profundidad de cola no es siempre uno o cero.

En este ejemplo también se muestran instancias del temporizador que se restablece sin que se generen sucesos, por ejemplo, en el momento P2.

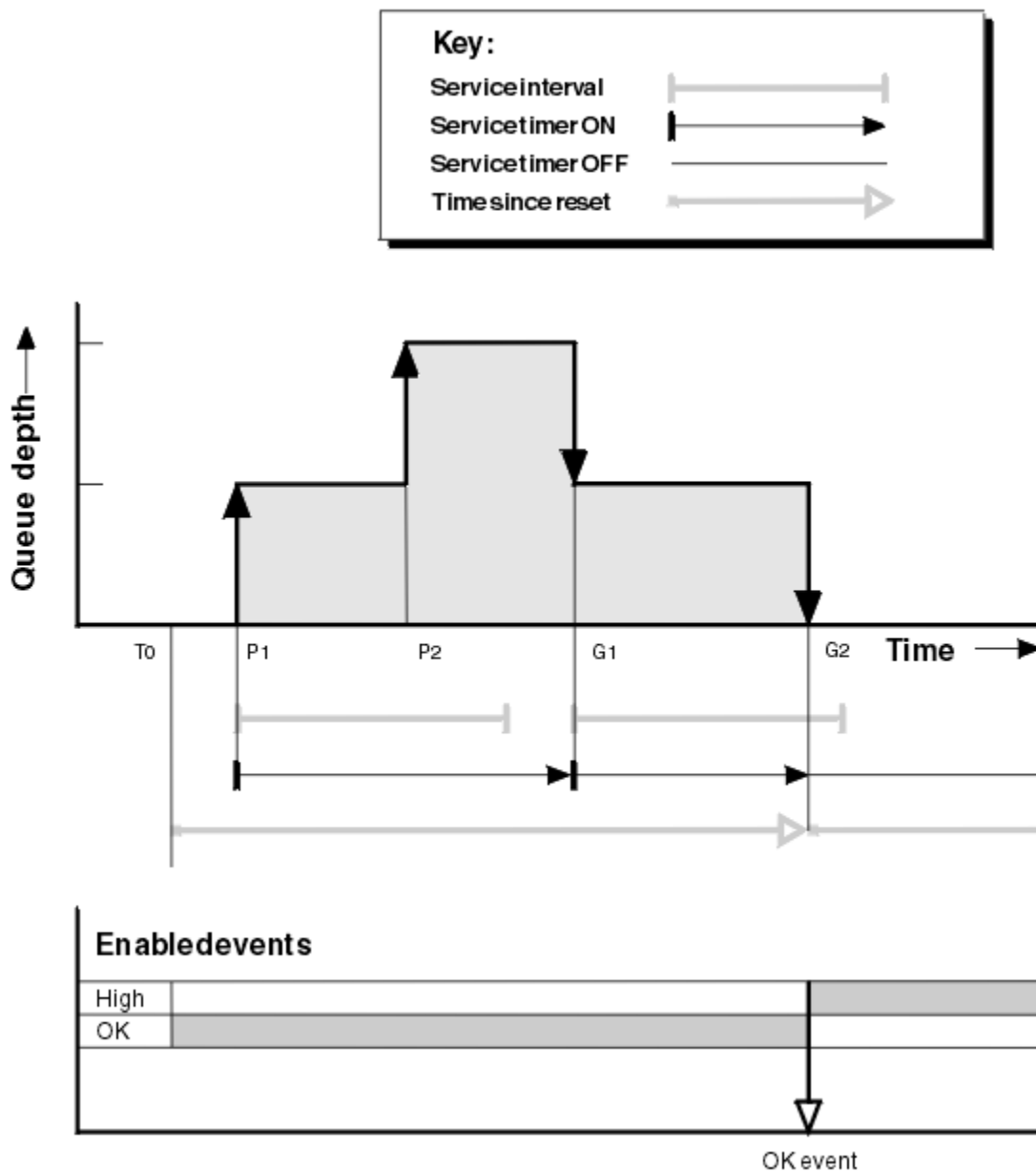


Figura 5. Sucesos de intervalo de servicio de cola: ejemplo 2

Comentario

En este ejemplo, los sucesos de tipo Correcto se han habilitado inicialmente, y en el momento T_0 se han restablecido las estadísticas de cola.

1. En P_1 , la primera transferencia inicia el temporizador de servicio.
2. En P_2 , la segunda transferencia no genera ningún suceso, debido a que una transferencia no puede provocar un suceso de tipo Correcto.
3. En G_1 , se ha superado el intervalo de servicio y, por tanto, no se genera ningún suceso de tipo Correcto. No obstante, la llamada MQGET hace que se restablezca el temporizador de servicio.
4. En G_2 , la segunda obtención se lleva a cabo dentro del intervalo de servicio, y esta vez se genera un suceso de tipo Correcto. El gestor de colas restablece el atributo de control de sucesos, de forma que:
 - a. El suceso de tipo Alto se habilita automáticamente.
 - b. El suceso de tipo Correcto se inhabilita.

Puesto que la cola ahora está vacía, el temporizador de servicio pasa a un estado OFF.

Resumen de estadísticas de sucesos

En la [Tabla 6](#) en la [página 34](#) se resumen las estadísticas de sucesos de este ejemplo.

<i>Tabla 6. Resumen de estadísticas de sucesos del ejemplo 2</i>	
Atributo	Suceso 2
Hora del suceso	T(G2)
Tipo de suceso	Correcto
TimeSinceReset	T(G2) - T(0)
HighQDepth	2
MsgEnqCount	2
MsgDeqCount	2

Sucesos de intervalo de servicio de cola: ejemplo 3

Una secuencia de llamadas MQGET y MQPUT, que es más esporádica que en los ejemplos anteriores.

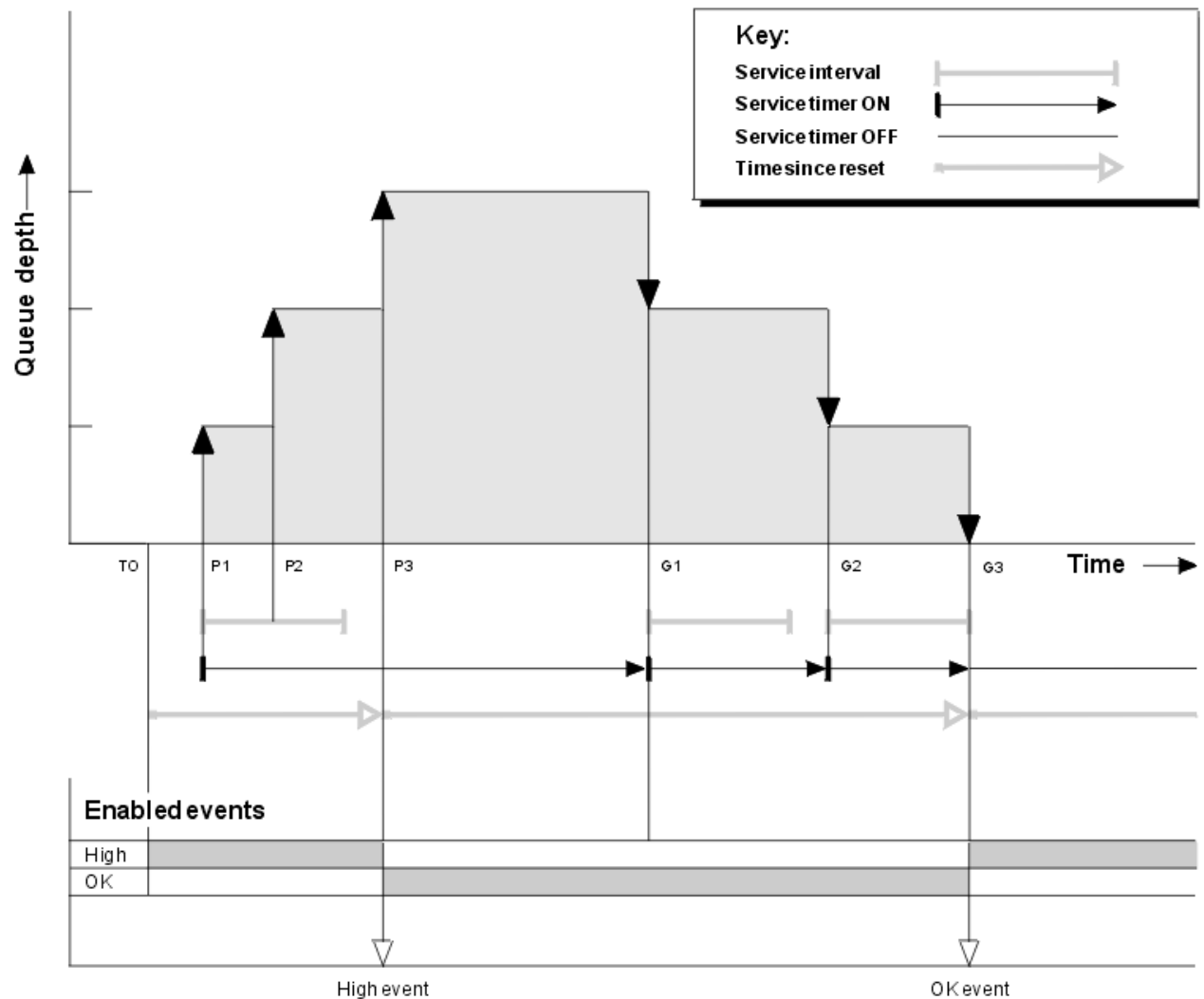


Figura 6. Sucesos de intervalo de servicio de cola: ejemplo 3

Comentario

1. En el momento T(0), se restablecen las estadísticas de la cola, y se habilitan los sucesos de intervalo de servicio de cola Alto.
2. En P1, la primera transferencia inicia el temporizador de servicio.
3. En P2, la segunda transferencia aumenta la profundidad de cola hasta dos. Aquí no se genera ningún suceso de tipo Alto porque no se ha superado el intervalo de tiempo de servicio.
4. En P3, la tercera transferencia hace que se genere un suceso de tipo Alto. (El temporizador ha excedido el intervalo de servicio.) El temporizador no se restablece, debido a que la profundidad de cola no era cero antes de efectuarse la transferencia. No obstante, los sucesos de tipo Correcto están habilitados.
5. En G1, la llamada MQGET no genera ningún suceso porque se ha superado el intervalo de servicio y los sucesos de tipo Correcto están habilitados. No obstante, la llamada MQGET sí restablece el temporizador de servicio.
6. En G2, la llamada MQGET no genera ningún suceso porque se ha superado el intervalo de servicio y los sucesos de tipo Correcto están habilitados. De nuevo, la llamada MQGET restablece el temporizador de servicio.
7. En G3, la tercera vacía la cola y el temporizador de servicio es igual al intervalo del servicio. Por tanto, se genera un suceso de tipo Correcto. El temporizador de servicio se restablece, y se habilitan los sucesos de tipo Alto. La llamada MQGET vacía la cola, y esta acción coloca el temporizador en el estado OFF.

Resumen de estadísticas de sucesos

En la [Tabla 7](#) en la [página 35](#) se resumen las estadísticas de sucesos de este ejemplo.

Atributo	Suceso 1	Suceso 2
Hora del suceso	T(P3)	T(G3)
Tipo de suceso	Alto	Correcto
TimeSinceReset	T(P3) - T(0)	T(G3) - T(P3)
HighQDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

Sucesos de profundidad de cola

Los sucesos de profundidad de cola están relacionados con la profundidad de cola, es decir, el número de mensajes que hay en la cola.

En las aplicaciones de IBM MQ, las colas no deben llenarse. Si se llenan, las aplicaciones ya no pueden transferir mensajes a la cola que se especifica. Aunque el mensaje no se pierde si esto se produce, una cola llena puede provocar unos inconvenientes considerables. El número de mensajes puede aumentar en una cola, si los mensajes se transfieren a la cola con más rapidez que la rapidez a la que pueden recogerlos las aplicaciones que los procesan.

La solución a este problema depende de las circunstancias particulares, pero podría implicar:

- Desviar algunos mensajes a otra cola.
- Iniciar nuevas aplicaciones para poder recoger más mensajes desde fuera de la cola.
- Detener el tráfico de mensajes que no sea esencial.
- Aumentar la profundidad de la cola para superar un máximo transitorio.

El aviso anticipado de que pueden surgir problemas hace que resulte más fácil tomar medidas preventivas. Con este fin, IBM MQ proporciona los siguientes sucesos de profundidad de cola:

Sucesos Cola de profundidad alta

Indican que la profundidad de cola ha aumentado hasta un umbral predefinido denominado el límite de profundidad de cola alta.

Sucesos Profundidad de cola baja

Indican que la profundidad de cola ha descendido hasta un umbral predefinido denominado el límite de profundidad de cola baja.

Sucesos Cola llena

Indican que la cola ha alcanzado su profundidad máxima, es decir, la cola está llena.

Un suceso de cola llena se genera cuando una aplicación intenta transferir un mensaje a una cola que ya ha alcanzado su profundidad máxima. Los sucesos de profundidad de cola alta dan un aviso anticipado de que una cola se está llenando. Esto significa que, tras haber recibido este suceso, el administrador del sistema debe realizar alguna acción preventiva. Puede configurar el gestor de colas de manera que, si la acción preventiva resulta satisfactoria, y la profundidad de cola desciende hasta un nivel más seguro, el gestor de colas genera un suceso de profundidad de cola baja.

En el primer ejemplo de suceso de profundidad de cola se ilustra el efecto de la presunta acción que impide que la cola se llene.

Conceptos relacionados

[“Ejemplos de sucesos de profundidad de cola” en la página 39](#)

Utilice estos ejemplos para comprender la información que puede obtener de los sucesos de profundidad de cola.

Referencia relacionada

[Cola llena](#)

[Profundidad de cola alta](#)



[Profundidad de cola baja](#)

Habilitación de sucesos de profundidad de cola

Para configurar una cola para cualquiera de los sucesos de profundidad de cola, debe establecer los atributos de gestor de colas y de cola pertinentes.

Acerca de esta tarea

De forma predeterminada, están inhabilitados todos los sucesos de profundidad de cola. Cuando están habilitados, los sucesos de profundidad de cola se generan tal como se indica a continuación:

- Un suceso de profundidad de cola alta se genera cuando se transfiere un mensaje a la cola, lo que provoca que la profundidad de la cola sea mayor o igual que el valor establecido para **QDepthHighLimit**.
 - Un suceso de profundidad de cola baja habilita automáticamente un suceso de profundidad de cola alta en la misma cola.
 - Un suceso de profundidad de cola alta habilita automáticamente tanto un suceso de profundidad de cola baja como un suceso de cola llena en la misma cola.
-  **Multi** Un suceso de profundidad de cola baja se genera cuando una operación GET elimina un mensaje una cola, lo que provoca que la profundidad de la cola sea menor o igual que el valor establecido para **QDepthLowLimit**.
-  **z/OS** Un suceso de profundidad de cola baja se genera cuando una operación GET elimina un mensaje una cola, o lo habría eliminado si no hubiera caducado, lo que provoca que la profundidad de la cola sea menor o igual que el valor establecido para **QDepthLowLimit**.
 - Un suceso de profundidad de cola alta o un suceso de cola llena habilitan automáticamente un suceso de profundidad de cola baja en la misma cola.

- Un suceso de profundidad de cola baja habilita automáticamente tanto un suceso de profundidad de cola alta como un suceso de cola llena en la misma cola.
- Un suceso de cola llena se genera cuando una aplicación no puede transferir un mensaje a una cola porque la cola está llena.
 - Un suceso de profundidad de cola alta o un suceso de profundidad de cola baja habilitan automáticamente un suceso de cola llena en la misma cola.
 - Un suceso de cola llena habilita automáticamente un suceso de profundidad de cola baja en la misma cola.

Efectúe los pasos siguientes para configurar una cola para cualquiera de los sucesos de profundidad de cola:

Procedimiento

1. Habilite los sucesos de rendimiento en el gestor de colas, mediante el atributo de gestor de colas **PERFMEV**.
Los sucesos van a la cola SYSTEM.ADMIN.PERFM.EVENT.
2. Establezca uno de los atributos siguientes para habilitar el suceso en la cola pertinente:
 - *QDepthHighEvent* (**QDPHIEV** en MQSC)
 - *QDepthLowEvent* (**QDPLOEV** en MQSC)
 - *QDepthMaxEvent* (**QDPMAXEV** en MQSC)
3. Opcional: Para establecer los límites, asigne los atributos siguientes, como el porcentaje máximo de la profundidad de cola:
 - *QDepthHighLimit* (**QDEPTHHI** en MQSC)
 - *QDepthLowLimit* (**QDEPTHLO** en MQSC)

Restricción: **QDEPTHHI** no debe ser menor que **QDEPTHLO**.

Si **QDEPTHHI** es igual que **QDEPTHLO**, se genera un mensaje de suceso cada vez que la profundidad de cola sobrepasa el valor en cualquier dirección, porque el umbral superior se habilita cuando la profundidad de cola está por debajo del valor, y se habilita el umbral inferior cuando la profundidad está por encima del valor.

Resultados

Nota:

Multi Un suceso de profundidad de cola baja no se genera cuando una operación GET elimina mensajes caducados de una cola, lo que provoca que la profundidad de la cola sea menor o igual que el valor establecido para **QDepthLowLimit**. IBM MQ genera el mensaje de suceso de profundidad de cola baja sólo durante una operación GET satisfactoria. Por lo tanto, cuando los mensajes caducados se eliminan de la cola, no se genera un mensaje de suceso de profundidad de cola baja. Además, después de la eliminación de estos mensajes caducados de la cola, *QDepthHighEvent* y *QDepthLowEvent* no se restablecen.

z/OS IBM MQ genera el suceso de profundidad de cola baja durante una operación GET destructiva satisfactoria, o una operación GET destructiva que habría sido satisfactoria si un mensaje coincidente no hubiera caducado. Por lo tanto, cuando los mensajes caducados se eliminan de una cola durante el proceso general en segundo plano, no se genera ningún mensaje de suceso de profundidad de cola baja. Además, después de la eliminación de estos mensajes caducados de la cola durante el proceso general en segundo plano, *QDepthHighEvent* y *QDepthLowEvent* no se restablecen. Para obtener más información sobre el proceso de mensajes caducados, consulte [Ajuste del gestor de colas en IBM MQ for z/OS](#).

Ejemplo

Para habilitar los sucesos de profundidad de cola alta en la cola MYQUEUE con un límite establecido en el 80%, utilice los mandatos MQSC siguientes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

Para habilitar los sucesos de profundidad de cola baja en la cola MYQUEUE con un límite establecido en el 20%, utilice los mandatos MQSC siguientes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHLO(20) QDPLOEV(ENABLED)
```

Para habilitar los sucesos de cola llena en la cola MYQUEUE, utilice los mandatos MQSC siguientes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDPMAXEV(ENABLED)
```

Shared queues and queue depth events on z/OS

On IBM MQ for z/OS, event monitoring is more straightforward for an application that uses shared queues if all the queue managers in the queue sharing group have the same setting for the **PERFMEV** attribute.

When a queue depth event occurs on a shared queue, and the queue manager attribute **PERFMEV** is set to **ENABLED**, the queue managers in the queue sharing group produce an event message. If **PERFMEV** is set to **DISABLED** on some of the queue managers, event messages are not produced by those queue managers, making event monitoring from an application more difficult. For more straightforward monitoring, give each queue manager the same setting for the **PERFMEV** attribute.

This event message that each queue manager generates represents its individual usage of the shared queue. If a queue manager performs no activity on the shared queue, various values in the event message are null or zero. You can use null event messages as follows:

- Ensure that each active queue manager in a queue sharing group generates one event message
- Highlight cases of no activity on a shared queue for the queue manager that produced the event message

Coordinating queue manager

When a queue manager issues a queue depth event, it updates the shared queue object definition to toggle the active performance event attributes. For example, depending on the definition of the queue attributes, a Queue Depth High event enables a Queue Depth Low and a Queue Full event. After updating the shared queue object successfully, the queue manager that detected the performance event initially becomes the *coordinating queue manager*.

If enabled for performance events, the coordinating queue manager performs the following actions:

1. Issues an event message that captures all shared queue performance data it has gathered since the last time an event message was created, or since the queue statistics were last reset. The message descriptor (MQMD) of this message contains a unique correlation identifier (*CorrelId*) created by the coordinating queue manager.
2. Broadcasts to all other *active* queue managers in the same queue sharing group to request the production of an event message for the shared queue. The broadcast contains the correlation identifier created by the coordinating queue manager for the set of event messages.

Having received a request from the coordinating queue manager, if there is an active queue manager in the queue sharing group that is enabled for performance events, that active queue manager issues an event message for the shared queue. The event message that is issued contains information about all the

operations performed by the receiving (active) queue manager since the last time an event message was created, or since the statistics were last reset. The message descriptor (MQMD) of this event message contains the unique correlation identifier (*CorrelId*) specified by the coordinating queue manager.

When performance events occur on a shared queue, *n* event messages are produced, where *n* is a number from 1 to the number of active queue managers in the queue sharing group. Each event message contains data that relates to the shared queue activity for the queue manager that generated the event message.

Differences between shared and nonshared queues

Enabling queue depth events on shared queues differs from enabling them on nonshared queues. A key difference is that events are enabled for shared queues even if **PERFMEV** is DISABLED on the queue manager. This is not the case for nonshared queues.

Consider the following example, which illustrates this difference:

- QM1 is a queue manager with *PerformanceEvent* (**PERFMEV** in MQSC) set to DISABLED.
- SQ1 is a shared queue with **QSGDISP** set to (SHARED) QLOCAL in MQSC.
- LQ1 is a nonshared queue with **QSGDISP** set to (QMGR) QLOCAL in MQSC.

Both queues have the following attributes set on their definitions:

- **QDPHIEV (ENABLED)**
- **QDPLOEV (DISABLED)**
- **QDPMAXEV (DISABLED)**

If messages are placed on both queues so that the depth meets or exceeds the **QDEPTHHI** threshold, the **QDPHIEV** value on SQ1 switches to DISABLED. Also, **QDPLOEV** and **QDPMAXEV** are switched to ENABLED. SQ1's attributes are automatically switched for each performance event at the time the event criteria are met.

In contrast the attributes for LQ1 remain unchanged until **PERFMEV** on the queue manager is ENABLED. This means that if, for example, the queue manager's **PERFMEV** attribute is ENABLED, DISABLED and then set to ENABLED again, the performance event settings on shared queues might not be consistent with those of nonshared queues, even though they might have initially been the same.

Ejemplos de sucesos de profundidad de cola

Utilice estos ejemplos para comprender la información que puede obtener de los sucesos de profundidad de cola.

En el primer ejemplo se proporciona una ilustración básica de los sucesos de profundidad de cola. El segundo ejemplo es más amplio, pero los principios son los mismos que para el primero. En ambos ejemplos se utiliza la misma definición de cola, tal como se indica a continuación:

La cola, MYQUEUE1, tiene una profundidad máxima de 1000 mensajes. El límite de profundidad de cola alta es del 80%, y el límite de profundidad de cola baja es del 20%. Inicialmente, los sucesos de profundidad de cola alta están habilitados, mientras que el resto de los sucesos de profundidad de cola están inhabilitados.

Los mandatos de IBM MQ (MQSC) para configurar esta cola son:

```
ALTER QMGR PERFMEV(ENABLED)

DEFINE QLOCAL('MYQUEUE1') MAXDEPTH(1000) QDPMAXEV(DISABLED) QDEPTHHI(80)
QDPHIEV(ENABLED) QDEPTHLO(20) QDPLOEV(DISABLED)
```

Conceptos relacionados

[“Sucesos de profundidad de cola” en la página 35](#)

Los sucesos de profundidad de cola están relacionados con la profundidad de cola, es decir, el número de mensajes que hay en la cola.

Tareas relacionadas

“Habilitación de sucesos de profundidad de cola” en la página 36

Para configurar una cola para cualquiera de los sucesos de profundidad de cola, debe establecer los atributos de gestor de colas y de cola pertinentes.

Referencia relacionada

Mandatos MQSC

Sucesos de profundidad de cola: ejemplo 1

Una secuencia básica de los sucesos de profundidad de cola.

En la Figura 7 en la página 40 se muestra la variación de la profundidad de cola en relación al tiempo.

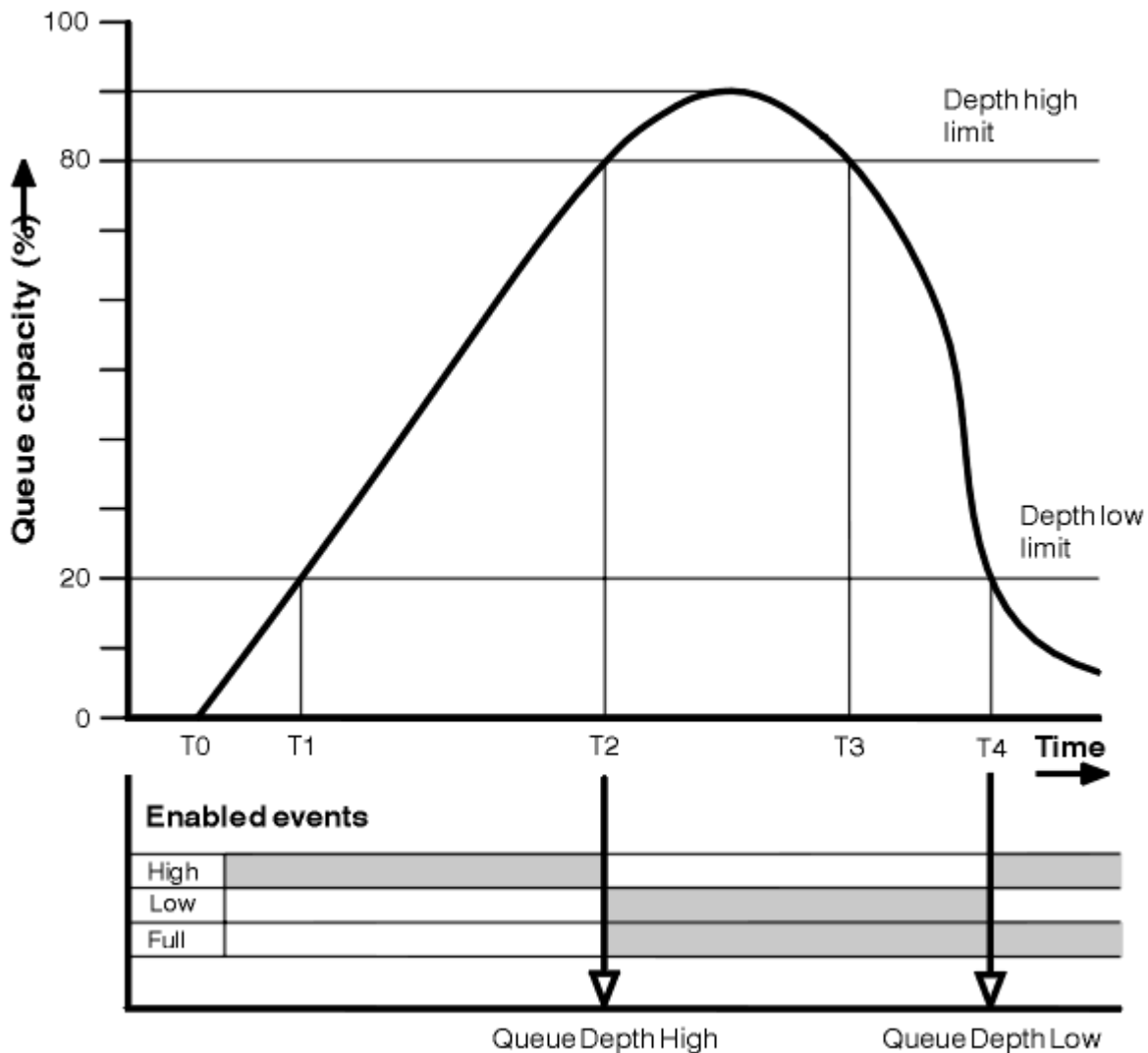


Figura 7. Sucesos de profundidad de cola (1)

Comentario

1. En T(1), la profundidad de la cola aumenta (más llamadas MQPUT que llamadas MQGET) y cruza el límite inferior de la profundidad de cola. En este momento no se genera ningún suceso.
2. La profundidad de cola continúa en aumento hasta T(2), cuando se alcanza el límite de profundidad alta (80%) y se genera un suceso de cola de profundidad alta.

Esto habilita tanto los sucesos Cola llena como los sucesos Profundidad de cola baja.

3. Las (presuntas) acciones preventivas promovidas por el suceso evitan que la cola se llene. En el momento T(3), se ha alcanzado de nuevo el límite de profundidad de cola alta, esta vez desde arriba. En este momento no se genera ningún suceso.
 4. La profundidad de cola continúa decreciendo hasta T(4), cuando se alcanza el límite de profundidad baja (20%) y se genera un suceso de cola de profundidad baja.
- Esto habilita tanto los sucesos Cola llena como los sucesos Cola de profundidad alta.

Resumen de estadísticas de sucesos

En la [Tabla 8 en la página 41](#) se resumen las estadísticas de sucesos de cola y en la [Tabla 9 en la página 41](#) se resumen los sucesos que están habilitados.

Tabla 8. Resumen de estadísticas de sucesos de los sucesos de profundidad de cola (ejemplo 1)

Estadísticas de suceso	Suceso 2	Suceso 4
Hora del suceso	T(2)	T(4)
Tipo de suceso	Profundidad de cola alta	Profundidad de cola baja
TimeSinceReset	T(2) - T(0)	T(4) - T(2)
HighQDepth (Profundidad de cola máxima desde el restablecimiento)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

Tabla 9. Resumen en el que se muestran qué sucesos están habilitados

Período de tiempo	Suceso Cola de profundidad alta	Suceso Profundidad de cola baja	Suceso Cola llena
Antes de T(1)	ENABLED	-	-
T(1) a T(2)	ENABLED	-	-
T(2) a T(3)	-	ENABLED	ENABLED
T(3) a T(4)	-	ENABLED	ENABLED
Después de T(4)	ENABLED	-	ENABLED

Sucesos de profundidad de cola: ejemplo 2

Una secuencia más amplia de los sucesos de profundidad de cola.

En la [Figura 8 en la página 42](#) se muestra la variación de la profundidad de cola en relación al tiempo.

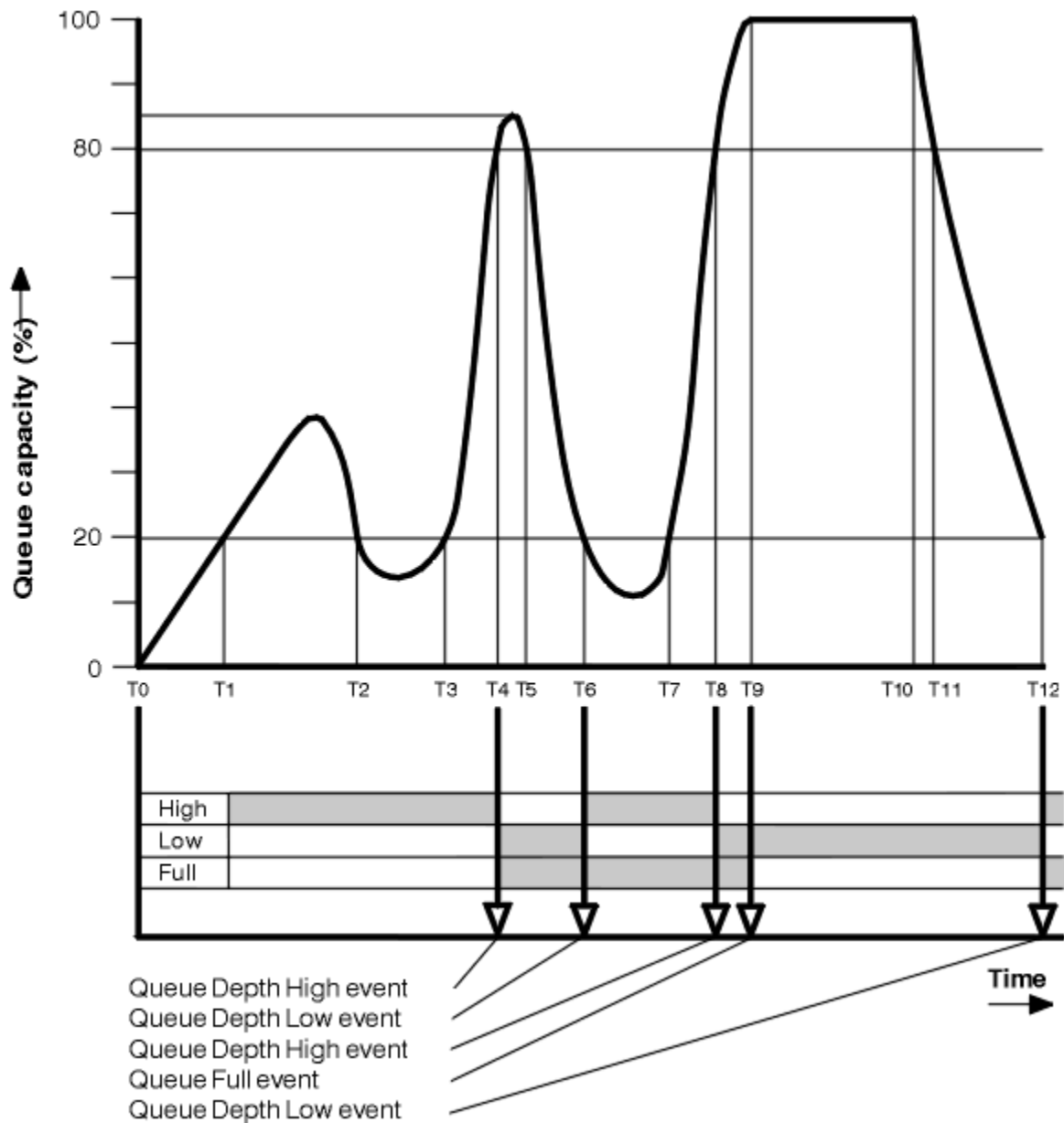


Figura 8. Sucesos de profundidad de cola (2)

Comentario

1. No se genera ningún suceso Profundidad de cola baja en los momentos siguientes:
 - T(1) (Profundidad de cola en aumento, y no habilitada)
 - T(2) (No habilitada)
 - T(3) (Profundidad de cola en aumento, y no habilitada)
2. En T(4) se produce un suceso Cola de profundidad alta. Esto habilita tanto los sucesos Cola llena como los sucesos Profundidad de cola baja.
3. En T(9) se produce un suceso Cola llena **después** del primer mensaje que no se pueda transferir a la cola porque se haya llenado ésta.
4. En T(12) se produce un suceso Profundidad de cola baja.

Resumen de estadísticas de sucesos

En la [Tabla 10](#) en la [página 43](#) se resumen las estadísticas de sucesos de cola y en la [Tabla 11](#) en la [página 43](#) se resumen los sucesos que están habilitados, en momentos diferentes para este ejemplo.

Tabla 10. Resumen de estadísticas de sucesos de los sucesos de profundidad de cola (ejemplo 2)

Estadísticas de suceso	Suceso 4	Suceso 6	Suceso 8	Suceso 9	Suceso 12
Hora del suceso	T(4)	T(6)	T(8)	T(9)	T(12)
Tipo de suceso	Profundidad de cola alta	Profundidad de cola baja	Profundidad de cola alta	Cola llena	Profundidad de cola baja
TimeSinceReset	T(4) - T(0)	T(6) - T(4)	T(8) - T(6)	T(9) - T(8)	T(12) - T(9)
HighQDepth	800	855	800	1000	1000
MsgEnqCount	1645	311	1377	324	221
MsgDeqCount	845	911	777	124	1021

Tabla 11. Resumen en el que se muestran qué sucesos están habilitados

Período de tiempo	Suceso Cola de profundidad alta	Suceso Profundidad de cola baja	Suceso Cola llena
T(0) a T(4)	ENABLED	-	-
T(4) a T(6)	-	ENABLED	ENABLED
T(6) a T(8)	ENABLED	-	ENABLED
T(8) a T(9)	-	ENABLED	ENABLED
T(9) a T(12)	-	ENABLED	-
Después de T(12)	ENABLED	-	ENABLED

Nota: Los sucesos están fuera del punto de sincronismo. Por tanto, podría tener una cola vacía, rellenarla provocando un suceso y, a continuación, retrotraer todos los mensajes que estén bajo el control de un gestor de punto de sincronismo. No obstante, la habilitación de sucesos se ha establecido automáticamente, de modo que la próxima vez que se llene la cola, no se generará ningún suceso.

Sucesos de configuración

Los sucesos de configuración son notificaciones que se generan cuando se crea un objeto, se cambia o se suprime y también se pueden generar mediante solicitudes explícitas.

Consulte [“Control de sucesos de configuración, mandato y registrador”](#) en la [página 22](#) para obtener información sobre cómo activar los sucesos.

Los sucesos de configuración le notifican sobre los atributos de un objeto. Existen cuatro tipos de suceso de configuración:

- Sucesos de creación de objeto
- Sucesos de cambio de objeto
- Sucesos de supresión de objeto
- Sucesos de renovación de objeto

Los datos de suceso contienen la información siguiente:

Información de origen

consta del gestor de colas desde el que ha efectuado el cambio, el ID del usuario que haya realizado el cambio, y cómo se efectuó el cambio; por ejemplo, mediante un mandato de consola.

Información de contexto

una réplica de la información de contexto de los datos de mensaje del mensaje de mandato.

La información de contexto se incluye en los datos del suceso sólo cuando el mandato se ha especificado como un mensaje en la cola SYSTEM.COMMAND.INPUT.

Identidad de objeto

consta del nombre, tipo y disposición del objeto.

Atributos de objetos

constan de los valores de todos los atributos del objeto.

El suceso es un mensaje en formato PCF. Si desea ver más información, consulte:

- [Cambiar objeto](#)
- [Crear objeto](#)
- [Suprimir objeto](#)
- [Renovar objeto](#)

En el caso de los sucesos de cambio de objeto, se genera un mensaje antes y un mensaje después. Tienen los campos siguientes en común, para que pueda identificar los mensajes coincidentes:

- El mismo ID de correlación
- En la cabecera PCF:
 - La cabecera del mensaje anterior tiene `MsgSeqNumber': 1, 'Control': 'NOT_LAST'`
 - La cabecera de mensaje posterior tiene `MsgSeqNumber': 2, 'Control': 'LAST'`

Todos los mensajes de suceso de configuración que se generan se colocan en la cola SYSTEM.ADMIN.CONFIG.EVENT.

Conceptos relacionados

[“Sucesos de configuración” en la página 17](#)

Los sucesos de configuración se generan cuando se solicita explícitamente un suceso de configuración, o se generan automáticamente al crear, modificar o suprimir un objeto.

Referencia relacionada

[Crear objeto](#)

[Cambiar objeto](#)

[Suprimir objeto](#)

[Renovar objeto](#)

[“Tipos de suceso” en la página 12](#)

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

Configuración de la generación de sucesos

Utilice esta página para ver los mandatos que provocan que se generen sucesos de configuración, y comprender las circunstancias en las que no se generan dichos sucesos.

Un mensaje de suceso de configuración se transfiere a la cola de sucesos de configuración cuando el atributo del gestor de colas CONFIGEV es ENABLED y

- se emite cualquiera de los mandatos siguientes, o su equivalente PCF:
 - DELETE AUTHINFO
 - DELETE CFSTRUCT
 - DELETE CHANNEL
 - DELETE NAMELIST

- DELETE PROCESS
- DELETE QMODEL/QALIAS/QREMOTE
- DELETE STGCLASS
- DELETE TOPIC
- REFRESH QMGR
- se emite cualquiera de los mandatos siguientes, o su equivalente PCF, aunque no se haya efectuado ningún cambio en el objeto:
 - DEFINE/ALTER AUTHINFO
 - DEFINE/ALTER CFSTRUCT
 - DEFINE/ALTER CHANNEL
 - DEFINE/ALTER NAMELIST
 - DEFINE/ALTER PROCESS
 - DEFINE/ALTER QMODEL/QALIAS/QREMOTE
 - DEFINE/ALTER STGCLASS
 - DEFINE/ALTER TOPIC
 - DEFINE MAXSMSGS
 - SET CHLAUTH
 - ALTER QMGR, a menos que el atributo CONFIGEV se haya establecido en DISABLED y no se cambie por ENABLED
- se emite cualquiera de los siguientes mandatos, o su equivalente PCF, para una cola local que no sea dinámica temporal, aunque no se haya efectuado ningún cambio en la cola.
 - DELETE QLOCAL
 - DEFINE/ALTER QLOCAL
- se emite llamada MQSET, que no sea para una cola dinámica temporal, aunque no se haya efectuado ningún cambio en el objeto.

Cuándo no se generan sucesos de configuración

Los mensajes de suceso de configuración no se generan en las circunstancias siguientes:

- Cuando un mandato o una llamada MQSET falla.
- Cuando un gestor de colas detecta un error al tratar de transferir un suceso de configuración a la cola de sucesos, en cuyo caso el mandato o la llamada MQSET finaliza, pero no se genera ningún mensaje de suceso.
- Si se trata de una cola dinámica temporal.
- Cuando se realizan cambios internos en el atributo de cola TRIGGER.
- Para la cola de sucesos de configuración SYSTEM.ADMIN.CONFIG.EVENT, excepto para el mandato QMGR REFRESH.
- Para los mandatos QMGR REFRESH/RESET CLUSTER y RESUME/SUSPEND, que provocan cambios de agrupación en clúster.
- Al crear o suprimir un gestor de colas.

Conceptos relacionados

Introducción a los Formatos de mandato programable

“Sucesos de configuración” en la página 43

Los sucesos de configuración son notificaciones que se generan cuando se crea un objeto, se cambia o se suprime y también se pueden generar mediante solicitudes explícitas.

Referencia relacionada

Mandatos MQSC

[MQSET - Establecer atributos de objeto](#)

[MQSET - Establecer atributos de objeto](#)

Uso de sucesos de configuración

Utilice esta página para ver cómo puede utilizar los sucesos de configuración para obtener información sobre el sistema, y para comprender los factores, tales como CMDSCOPE, que puede afectar al uso de los sucesos de configuración.

Puede utilizar los sucesos de configuración para los fines siguientes:

1. Producir y mantener un repositorio de configuración central, desde el cual se puedan producir informes y generar información sobre la estructura del sistema.
2. Generar un seguimiento de auditoría. Por ejemplo, si un objeto se cambia de forma inesperada, se puede almacenar la información sobre quién y cuándo se efectuó la modificación.

Esto puede resultar especialmente útil cuando también están habilitados los sucesos de mandato. Si un mandato MQSC o PCF provoca que se generen un suceso de configuración y uno de mandato, ambos mensajes de suceso compartirán el mismo identificador de correlación en su descriptor de mensaje.

Para una llamada MQSET, o cualquiera de los mandatos siguientes:

- DEFINE objeto
- ALTER objeto
- DELETE objeto

Si el atributo de gestor de colas CONFIGEV está habilitado, pero el mensaje de suceso de configuración no se puede colocar en la cola de sucesos de configuración, por ejemplo no se ha definido la cola de sucesos, el mandato o la llamada MQSET se ejecuta de todos modos.

Efectos de CMDSCOPE

En el caso de los mandatos en los que se utilice CMDSCOPE, se generará el mensaje, o mensajes, de suceso de configuración en el gestor, o gestores, de colas en los que se ejecute el mandato, no en los que se introduzca éste. No obstante, toda la información de origen y de contexto de los datos de suceso serán relativos al mandato original tal como se haya especificado, aunque el mandato en que se utilice CMDSCOPE sea uno que haya generado el gestor de colas de origen.

En el caso que un grupo de compartición incluya gestores de colas que no tengan la versión actual, se generarán sucesos para cualquier mandato que se ejecute por medio de CMDSCOPE en un gestor de colas que se encuentre en la versión actual, pero no en los que se encuentren en una versión anterior. Esto sucede aunque el gestor de colas en el que se haya introducido el mandato se encuentre en la versión anterior aunque, en este caso, no haya información de contexto que se incluya en los datos de suceso.

Conceptos relacionados

[Introducción a los Formatos de mandato programable](#)

[“Sucesos de configuración” en la página 43](#)

Los sucesos de configuración son notificaciones que se generan cuando se crea un objeto, se cambia o se suprime y también se pueden generar mediante solicitudes explícitas.

Referencia relacionada

[MQSET - Establecer atributos de objeto](#)

[MQSET - Establecer atributos de objeto](#)

Suceso de configuración Renovar objeto

El suceso de configuración Renovar objeto es diferente de los demás sucesos de configuración, porque sólo se genera cuando se solicita explícitamente.

Los sucesos de creación, cambio y supresión los genera una llamada MQSET, o un mandato de cambio de un objeto, pero el suceso sólo se produce cuando lo solicita explícitamente el mandato MQSC, REFRESH QMGR, o su equivalente PCF.

El mandato REFRESH QMGR es diferente de todos los demás mandatos que generan sucesos de configuración. Todos los demás mandatos se aplican a un objeto particular y generan un único suceso de configuración para dicho objeto. El mandato REFRESH QMGR puede producir muchos mensajes de suceso de configuración que, potencialmente, representan todas las definiciones de objeto que un gestor de colas puede almacenar. Se genera un mensaje de suceso para cada objeto que se haya seleccionado.

El mandato REFRESH QMGR utiliza una combinación de tres criterios de selección para filtrar el número de objetos implicados:

- Nombre de objeto
- Tipo de objeto
- Intervalo de renovación

Si no especifica ninguno de los criterios de selección en el mandato REFRESH QMGR, se utilizan los valores predeterminados para cada criterio de selección y se genera un mensaje de suceso para cada definición de objeto de renovación de configuración que el gestor de colas haya almacenado. Esto puede ocasionar tiempos de proceso inaceptables y que se generen mensajes de suceso. Considere la posibilidad de especificar algunos criterios de selección.

El mandato REFRESH QMGR que genera los sucesos de renovación puede utilizarse en las situaciones siguientes:

- Cuando se desean datos de configuración sobre todos o algunos de los objetos de un sistema, independientemente de si los objetos se han manipulado recientemente, por ejemplo, cuando se habilitan por primera vez los sucesos de configuración.

Considere la posibilidad de utilizar varios mandatos, cada uno con una selección de objetos distinta, pero de tal modo que se incluyan todos.

- Si se ha producido un error en la cola SYSTEM.ADMIN.CONFIG.EVENT. En estas circunstancias, no se generan mensajes de suceso de configuración para los sucesos de creación, cambio ni supresión. Una vez corregido el error de la cola, se puede utilizar el mandato Renovar gestor de colas para solicitar la generación de los mensajes de suceso, que se perdieron mientras había un error en la cola. En esta situación, considere la posibilidad de establecer el intervalo de renovación en el tiempo durante el que la cola no estaba disponible.

Conceptos relacionados

[“Sucesos de configuración” en la página 43](#)

Los sucesos de configuración son notificaciones que se generan cuando se crea un objeto, se cambia o se suprime y también se pueden generar mediante solicitudes explícitas.

Referencia relacionada

[REFRESH QMGR](#)

[Renovar gestor de colas](#)

Sucesos de mandatos

Los sucesos de mandato son notificaciones que un mandato PCF o MQSC ha ejecutado de forma satisfactoria.

Los datos de suceso contienen la información siguiente:

Información de origen

consta del gestor de colas desde el que se ha emitido el mandato, el ID del usuario que lo haya emitido, y cómo se emitió el mismo, por ejemplo, mediante un mandato de consola.

Información de contexto

una réplica de la información de contexto de los datos de mensaje del mensaje de mandato. Si no se entra ningún mandato utilizando un mensaje, la información de contexto se omite.

La información de contexto se incluye en los datos del suceso sólo cuando el mandato se ha especificado como un mensaje en la cola SYSTEM.COMMAND.INPUT.

Información del mandato

el tipo de mandato que se ha emitido.

Datos de mandato

- para los mandatos PCF, una réplica de los datos de mandato.
- para los mandatos MQSC, el texto del mandato.

El formato de los datos de mandato no coincide necesariamente con el formato del mandato original. Por ejemplo, en [Multiplatforms](#), el formato de los datos de mandato siempre es el formato PCF, aunque la solicitud original sea un mandato MQSC.

Todos los mensajes de suceso de mandato que se generan se colocan en la cola SYSTEM.ADMIN.COMMAND.EVENT.

Referencia relacionada

Mandato

[“Tipos de suceso” en la página 12](#)

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

Generación de sucesos de mandatos

Utilice esta página para ver las situaciones que provocan que se generen los sucesos de mandato, y comprender las circunstancias en las que no se generan dichos sucesos.

Un mensaje de suceso de mandato se genera en las circunstancias siguientes:

- Cuando se especifica el atributo de gestor de colas CMDEV como ENABLED y se ejecuta de forma satisfactoria un mandato MQSC o PCF.
- Cuando se especifica el atributo de gestor de colas CMDEV como NODISPLAY, y cualquier mandato se ejecuta de forma satisfactoria, con la excepción de los mandatos DISPLAY (MQSC) e Inquire (PCF).
- Al ejecutar el mandato MQSC, ALTER QMGR, o el mandato PCF, Cambiar gestor de colas, y el atributo de gestor de colas CMDEV cumple una de las condiciones siguientes:
 - CMDEV no se especifica como DISABLED después del cambio
 - CMDEV no se ha especificado como DISABLED antes del cambio

Si un mandato se ejecuta en la cola de sucesos de mandato, SYSTEM.ADMIN.COMMAND.EVENT, se genera un suceso de mandato si la cola aún existe, y no se ha inhibido para las operaciones de tipo put.

Cuándo no se generan los sucesos de mandatos

Un mensaje de suceso de mandato no se genera en las circunstancias siguientes:

- Cuando un mandato falla.
- Cuando un gestor de colas detecta un error al tratar de transferir un suceso de mandato a la cola de sucesos, en cuyo caso el mandato se ejecuta de todos modos, pero no se genera ningún mensaje de suceso.
- Para el mandato MQSC REFRESH QMGR TYPE (EARLY).
- Para el mandato MQSC START QMGR MQSC.
- Para el mandato MQSC SUSPEND QMGR, si se especifica LOG el parámetro.
- Para el mandato MQSC RESUME QMGR, si se especifica LOG el parámetro.

Conceptos relacionados

[“Sucesos de mandatos” en la página 47](#)

Los sucesos de mandato son notificaciones que un mandato PCF o MQSC ha ejecutado de forma satisfactoria.

Referencia relacionada

[REFRESH QMGR](#)

[START QMGR](#)

[SUSPEND QMGR](#)

[RESUME QMGR](#)

[SUSPEND QMGR, RESUME QMGR y clústeres](#)

Uso de sucesos de mandato

Utilice esta página para ver cómo puede utilizar los sucesos de mandato para generar un seguimiento de auditoría de los mandatos que se hayan ejecutado.

Por ejemplo, si un objeto se cambia de forma inesperada, se puede almacenar la información sobre quién y cuándo se efectuó la modificación. Esto puede resultar especialmente útil cuando también están habilitados los sucesos de configuración. Si un mandato MQSC o PCF provoca que se generen un suceso de mandato y uno de configuración, ambos mensajes de suceso compartirán el mismo identificador de correlación en su descriptor de mensaje.

Si se genera un mensaje de suceso de mandato, pero no se puede colocar en la cola de sucesos de mandato, por ejemplo, si no se ha definido la cola de sucesos de mandato, el mandato para que se ha generado el suceso de mandato se ejecuta de todos modos.

Efectos de CMDSCOPE

En el caso de los mandatos en los que se utilice CMDSCOPE, se generará el mensaje, o mensajes, de suceso de mandato en el gestor, o gestores, de colas en los que se ejecute el mandato, no en los que se introduzca éste. No obstante, toda la información de origen y de contexto de los datos de suceso serán relativos al mandato original tal como se haya especificado, aunque el mandato en que se utilice CMDSCOPE sea uno que haya generado el gestor de colas de origen.

Conceptos relacionados

[“Sucesos de mandatos” en la página 47](#)

Los sucesos de mandato son notificaciones que un mandato PCF o MQSC ha ejecutado de forma satisfactoria.

[“Generación de sucesos de mandatos” en la página 48](#)

Utilice esta página para ver las situaciones que provocan que se generen los sucesos de mandato, y comprender las circunstancias en las que no se generan dichos sucesos.

Referencia relacionada

[Mandatos MQSC](#)

[Mandatos PCF y respuestas en grupos](#)

Multi

Sucesos del registrador de anotaciones

Los sucesos de registrador son notificaciones de que un gestor de colas ha empezado a grabar en una nueva extensión de registro **IBM i** o, en IBM i, un destinatario de diario. **z/OS** Los mensajes de sucesos de registrador no están disponibles con IBM MQ for z/OS.

Los datos de suceso contienen la información siguiente:

- El nombre de la extensión de registro actual.
- El nombre de la extensión de registro más antigua necesaria para poder llevar a cabo la recuperación de reinicio.
- El nombre de la extensión de registro más antigua necesaria para poder llevar a cabo la recuperación desde soporte de almacenamiento.
- El directorio en el que residen las extensiones de registro.
- El nombre de la extensión de registro más antigua para la que se requiere una notificación de archivado.

Se genera un suceso de registrador para la gestión de registros de archivado, es decir, cuando **ARCHLOG** cambia, y el valor **ARCHLOG** se incluye en el mensaje de suceso de registrador.

Consulte [DISPLAY QMSTATUS](#) para obtener más información sobre todos estos parámetros.

Puede utilizar los valores **CURRLOG** y **ARCHLOG** para determinar qué debe archivarse. Cuando **CURRLOG** cambia, los archivos con números menores que **CURRLOG** pueden enviarse para su archivado y, una vez completado el archivado para la extensión, debe invocar [SET LOG](#) para notificar al gestor de colas.

ARCHLOG es la extensión más antigua que necesita archivado. Cuando haya archivado dicha extensión, y haya llamado a [SET LOG](#) para notificar al gestor de colas que se ha archivado, el gestor de colas traslada **ARCHLOG** hasta la siguiente extensión de la secuencia. El gestor de colas planifica las extensiones que son más antiguas que **ARCHLOG**, para ser suprimidas o reutilizadas.

Una vez que se ha llamado a **SET LOG** para esta extensión, se emite un nuevo suceso, ya que **ARCHLOG** habrá cambiado, y necesita saber qué se debe archivar después del nuevo suceso.

Si por alguna razón el proceso de archivado falla y se produce una gran acumulación de notificaciones, el administrador puede emitir manualmente el mandato [RESET QMGR TYPE \(ARCHLOG\)](#). Esto notifica al gestor de colas que puede reutilizar o suprimir todas las extensiones más antiguas que, e incluyendo, la extensión que ha especificado.

Todos los mensajes de suceso de registrador que se generan se colocan en la cola de sucesos de registrador, SYSTEM.ADMIN.LOGGER.EVENT.

Referencia relacionada

[Registrador](#)

[“Tipos de suceso” en la página 12](#)

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

Generación de sucesos del registrador

Utilice esta página para ver las situaciones que provocan la generación de sucesos de registrador y comprender las circunstancias en las no se generan dichos sucesos.

Un mensaje de suceso del registrador de anotaciones se genera en las circunstancias siguientes:

- Cuando se ha especificado el atributo de gestor de colas LOGGERSV como ENABLED, y el gestor de colas empieza a grabar en una nueva extensión de registro o, en IBM i, un destinatario de diario.
- Cuando el atributo de gestor de colas LOGGERSV se ha especificado como ENABLED y se inicia el gestor de colas.
- Cuando el atributo de gestor de colas LOGGERSV se cambia de DISABLED a ENABLED.
- Cuando se notifica al atributo de gestor de colas LOGGERSV que se ha archivado el nombre de la extensión de registro más antigua, para la que el gestor de colas espera la notificación de archivado (ARCHLOG).

Consejo: Puede utilizar el mandato MQSC, RESET QMGR, para solicitar que un gestor de colas empiece a escribir en una nueva extensión de registro.

Cuándo no se generan los sucesos del registrador de anotaciones

Un mensaje de suceso de registrador no se genera en las circunstancias siguientes:

- Cuando un gestor de colas está configurado para utilizar el registro circular.

En este caso, el atributo de gestor de colas LOGGERSV se establece como DISABLED, y no se puede modificar.

- Cuando un gestor de colas detecta un error al intentar colocar un suceso de registrador en la cola de sucesos, en cuyo caso la acción que ha provocado el suceso se completa, pero no se genera ningún mensaje de suceso.

Conceptos relacionados

[“Sucesos del registrador de anotaciones” en la página 49](#)

Los sucesos de registrador son notificaciones de que un gestor de colas ha empezado a grabar en una nueva extensión de registro **IBM i** o, en IBM i, un destinatario de diario. **z/OS** Los mensajes de sucesos de registrador no están disponibles con IBM MQ for z/OS.

Referencia relacionada

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(entero con signo de 10 dígitos\)](#)

[RESET QMGR](#)

Multi *Uso del registrador de anotaciones*

Utilice esta página para ver cómo puede utilizar los sucesos de registrador para determinar las extensiones de registro que ya no sean necesarias para el reinicio del gestor de colas, o para poder llevar a cabo la recuperación desde soporte de almacenamiento.

Puede archivar las extensiones de registro superfluas en un soporte de almacenamiento como, por ejemplo, una cinta, a efectos de recuperación tras desastre, antes de eliminarlas del directorio de registro activo. La eliminación regular de las extensiones de registro superfluas logra que el uso de espacio de disco se mantenga al mínimo.

Si el atributo de gestor de colas LOGGEREV está habilitado, pero no se puede colocar un mensaje de suceso de registrador en la cola de sucesos de registrador, por ejemplo porque se no ha definido la cola de sucesos, la acción que ha provocado el suceso continúa de todos modos.

Conceptos relacionados

[“Sucesos del registrador de anotaciones” en la página 49](#)

Los sucesos de registrador son notificaciones de que un gestor de colas ha empezado a grabar en una nueva extensión de registro **IBM i** o, en IBM i, un destinatario de diario. **z/OS** Los mensajes de sucesos de registrador no están disponibles con IBM MQ for z/OS.

Referencia relacionada

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(entero con signo de 10 dígitos\)](#)

[“Generación de sucesos del registrador” en la página 50](#)

Utilice esta página para ver las situaciones que provocan la generación de sucesos de registrador y comprender las circunstancias en las no se generan dichos sucesos.

Multi *Programa C de ejemplo para supervisar la cola de sucesos del registrador*

Utilice esta página para ver un programa de ejemplo, escrito en C, que supervisa la cola de sucesos de registrador en busca de mensajes de suceso nuevos, lee dichos mensajes, y coloca su contenido en la salida estándar (stdout).

```
/*
/*****
/*
/* Program name: AMQSLOG0.C
/*
/*
/* Description: Sample C program to monitor the logger event queue and
/* display formatted message content to stdout when a logger
/* event occurs
/*
/* <copyright
/* notice="lm-source-program"
/* pids="5724-H72,"
/* years="2005, 2024"
/* crc="186943832" >
/* Licensed Materials - Property of IBM
/*
/*
/* 5724-H72,
/*
/*
/* (C) Copyright IBM Corp. 2005, 2024. All Rights Reserved.
/*
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/*
/* </copyright>
/*****
*/
```

```

/*
/* Function: AMQSLOG is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and displays the
/* formatted contents of the message to stdout.
/*
/*
/*****
/*
/* AMQSLOG has 1 parameter - the queue manager name (optional, if not
/* specified then the default queue manager is implied)
/*
/*
/*****

/*****
/* Includes
/*****
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include <cmqc.h>          /* MQI constants*/
#include <cmqcfc.h>       /* PCF constants*/

/*****
/* Constants
/*****

#define    MAX_MESSAGE_LENGTH    8000

typedef struct _ParmTableEntry
{
    MQLONG    ConstVal;
    PMQCHAR   Desc;
} ParmTableEntry;

ParmTableEntry ParmTable[] =
{
    {0, "",},
    {MQCA_Q_MGR_NAME, "Queue Manager Name"},
    {MQCMD_LOGGER_EVENT, "Logger Event Command"},
    {MQRC_LOGGER_STATUS, "Logger Status"},
    {MQCACF_ARCHIVE_LOG_EXTENT_NAME, "Archive Log Extent"},
    {MQCACF_CURRENT_LOG_EXTENT_NAME, "Current Log Extent"},
    {MQCACF_RESTART_LOG_EXTENT_NAME, "Restart Log Extent"},
    {MQCACF_MEDIA_LOG_EXTENT_NAME, "Media Log Extent"},
    {MQCACF_LOG_PATH, "Log Path"}
};

#if defined(MQ_64_BIT)
    #define Int32
#else
    #define Int32 "l"
#endif

/*****
/* Function prototypes
/*****

static void ProcessPCF(MQHCONN    hConn,
                      MQHOBJ     hEventQueue,
                      PMQCHAR    pBuffer);

static PMQCHAR ParmToString(MQLONG Parameter);

/*****
/* Function: main
/*****

int main(int argc, char * argv[])
{
    MQLONG    CompCode;
    MQLONG    Reason;
    MQHCONN   hConn = MQHC_UNUSABLE_HCONN;
    MQOD      ObjDesc = { MQOD_DEFAULT };
    MQCHAR    QMName[MQ_Q_MGR_NAME_LENGTH+1] = "";
    MQCHAR    LogEvQ[MQ_Q_NAME_LENGTH] = "SYSTEM.ADMIN.LOGGER.EVENT";
    MQHOBJ    hEventQueue = MQHO_UNUSABLE_HOBJ;
    PMQCHAR   pBuffer = NULL;

    printf("\n/*****/\n");
    printf("/* Sample Logger Event Monitor start */\n");
    printf("/* *****/\n");

```

```

/*****
/* Parse any command line options */
/*****
if (argc > 1)
{
    strncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
}

pBuffer = (PMQCHAR)malloc(MAX_MESSAGE_LENGTH);
if (pBuffer == NULL)
{
    printf("Can't allocate %d bytes\n", MAX_MESSAGE_LENGTH);
    goto MOD_EXIT;
}

/*****
/* Connect to the specified (or default) queue manager */
/*****
MQCONN(QMName,
        &hConn,
        &CompCode,
        &Reason);

if (Reason != MQRC_NONE)
{
    printf("MQCONN ended with reason code %" Int32 "d\n", Reason);
    goto MOD_EXIT;
}

/*****
/* Open the logger event queue for input */
/*****
strncpy(ObjDesc.ObjectQMgrName, QMName, MQ_Q_MGR_NAME_LENGTH);
strncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

MQOPEN(hConn,
        &ObjDesc,
        MQOO_INPUT_EXCLUSIVE,
        &hEventQueue,
        &CompCode,
        &Reason );

if (Reason != MQRC_NONE)
{
    printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %" Int32 "d\n",
            ObjDesc.ObjectQMgrName,
            ObjDesc.ObjectName,
            Reason);
    goto MOD_EXIT;
}
else
{
    /*****
    /* Start processing event messages */
    /*****
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:
if (pBuffer != NULL)
{
    free(pBuffer);
}

/*****
/* Close the logger event queue */
/*****
if (hEventQueue != MQHO_UNUSABLE_HOBJ)
{
    MQCLOSE(hConn, &hEventQueue, MQCO_NONE, &CompCode, &Reason);
}

/*****
/* Disconnect */
/*****
if (hConn != MQHC_UNUSABLE_HCONN)
{
    MQDISC(&hConn, &CompCode, &Reason);
}

return 0;

```

```

}

/*****
/* Function: ProcessPCF
/*****
/*
/* Input Parameters: Handle to queue manager connection
/* Handle to the opened logger event queue object
/* Pointer to a memory buffer to store the incoming PCF
/* message
/*
/* Output Parameters: None
/*
/* Logic: Wait for messages to appear on the logger event queue and display
/* their formatted contents.
/*
/*
/*****

static void ProcessPCF(MQHCONN hConn,
                     MQHOBJ hEventQueue,
                     PMQCHAR pBuffer)
{
    MQCFH * pCfh;
    MQCFST * pCfst;
    MQGMO Gmo = { MQGMO_DEFAULT };
    MQMD Mqmd = { MQMD_DEFAULT };
    PMQCHAR pPCFCmd;
    MQLONG CompCode = MQCC_OK;
    MQLONG Reason = MQRC_NONE;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    Gmo.Options = MQGMO_WAIT +
                 MQGMO_CONVERT +
                 MQGMO_FAIL_IF QUIESCING;
    Gmo.WaitInterval = MQWI_UNLIMITED; /* Set timeout value */

    /*****
    /* Process response Queue
    /*****
    while (Reason == MQRC_NONE)
    {
        memcpy(&Mqmd.MsgId, MQMI_NONE, sizeof(Mqmd.MsgId));
        memset(&Mqmd.CorrelId, 0, sizeof(Mqmd.CorrelId));

        MQGET( hConn,
              hEventQueue,
              &Mqmd,
              &Gmo,
              MAX_MESSAGE_LENGTH,
              pBuffer,
              &MsgLen,
              &CompCode,
              &Reason );

        if (Reason != MQRC_NONE)
        {
            switch(Reason)
            {
                case MQRC_NO_MSG_AVAILABLE:
                    printf("Timed out");
                    break;

                default:
                    printf("MQGET ended with reason code %" Int32 "d\n", Reason);
                    break;
            }
            goto MOD_EXIT;
        }

        /*****
        /* Only expect PCF event messages on this queue
        /*****
        if (memcmp(Mqmd.Format, MQFMT_EVENT, MQ_FORMAT_LENGTH))
        {
            printf("Unexpected message format '%8.8s' received\n", Mqmd.Format);
            continue;
        }

        /*****
        /* Build the output by parsing the received PCF message, first the
        /* header, then each of the parameters
        /*****

```

```

/*****
pCfh = (MQCFH *)pBuffer;

if (pCfh->Reason != MQRC_NONE)
{
printf("-----\n");
printf("Event Message Received\n");

Parm = ParmToString(pCfh->Command);
if (Parm != NULL)
{
printf("Command  :%s \n",Parm);
}
else
{
printf("Command  :%" Int32 "d \n",pCfh->Command);
}

printf("CompCode :%" Int32 "d\n" ,pCfh->CompCode);

Parm = ParmToString(pCfh->Reason);
if (Parm != NULL)
{
printf("Reason   :%s \n",Parm);
}
else
{
printf("Reason   :%" Int32 "d \n",pCfh->Reason);
}
}

pPCFCmd = (PMQCHAR) (pCfh+1);
printf("-----\n");
while(pCfh->ParameterCount-->0)
{
pCfst = (MQCFST *) pPCFCmd;
switch(pCfst->Type)
{
case MQCFT_STRING:
Parm = ParmToString(pCfst->Parameter);
if (Parm != NULL)
{
printf("%-32s",Parm);
}
else
{
printf("%-32" Int32 "d",pCfst->Parameter);
}

fwrite(pCfst->String, pCfst->StringLength, 1, stdout);
pPCFCmd += pCfst->StrucLength;
break;

default:
printf("Unrecognised datatype %" Int32 "d returned\n", pCfst->Type);
goto MOD_EXIT;
}
putchar('\n');
}
}

```

Salida de ejemplo

Esta aplicación genera el formato de salida siguiente:

```

/*****
/* Sample Logger Event Monitor start */
/*****
-----
Event Message Received
Command :Logger Event Command
CompCode :0
Reason  :Logger Status
-----
Queue Manager Name      CSIM

Current Log Extent      AMQA000001
Restart Log Extent     AMQA000001
Media Log Extent       AMQA000001

```

Conceptos relacionados

[“Uso del registrador de anotaciones” en la página 51](#)

Utilice esta página para ver cómo puede utilizar los sucesos de registrador para determinar las extensiones de registro que ya no sean necesarias para el reinicio del gestor de colas, o para poder llevar a cabo la recuperación desde soporte de almacenamiento.

[“Uso de sucesos de mandato” en la página 49](#)

Utilice esta página para ver cómo puede utilizar los sucesos de mandato para generar un seguimiento de auditoría de los mandatos que se hayan ejecutado.

Referencia relacionada

[“Generación de sucesos del registrador” en la página 50](#)

Utilice esta página para ver las situaciones que provocan la generación de sucesos de registrador y comprender las circunstancias en las que no se generan dichos sucesos.

Sucesos de configuración de autorización

Los sucesos de configuración de autorización se crean cuando se realiza un cambio desde cualquiera de las operaciones de control de seguridad a través de la línea de mandatos, MQSC, PCF o los mandatos de IBM i correspondientes.

Los datos de suceso contienen la información siguiente:

Información de origen

consta del gestor de colas desde el que ha efectuado el cambio, el ID del usuario que haya realizado el cambio, y cómo se efectuó el cambio; por ejemplo, mediante un mandato de consola.

Información de contexto

una réplica de la información de contexto de los datos de mensaje del mensaje de mandato.

La información de contexto se incluye en los datos del suceso cuando el mandato se ha entrado como un mensaje en la cola SYSTEM.ADMIN.COMMAND.QUEUE.

Identidad de registro de autorización

comprende el nombre de perfil y el tipo de objeto del registro de autorización.

Atributos de objetos

comprende los valores de todos los atributos en el registro de autorización.

En el caso de los sucesos de cambio de registro de autorización, se generan dos mensajes, uno con la información antes del cambio, el otro con la información después del cambio.

Cada mensaje de suceso que se genera se coloca en la cola SYSTEM.ADMIN.CONFIG.EVENT.

Referencia relacionada

[“Tipos de suceso” en la página 12](#)

Utilice esta página para ver los tipos de suceso de instrumentación sobre los que puede informar un gestor de colas, o la instancia de canal.

Generación de sucesos de configuración de autorización

Utilice esta página para ver las situaciones que causan que se generen sucesos de configuración de autorización y para comprender las circunstancias en las que los sucesos de configuración de autorización no se generan.

Los sucesos de configuración de autorización le notifican acerca de cambios en los atributos de un registro de autorización. Existen tres tipos de suceso de configuración de autorización:

- [Cambiar registro de autorización](#)
- [Suprimir registro de autorización](#)
- [Renovar registro de autorización](#)

Un mensaje de suceso de autorización se coloca en la cola de sucesos de configuración cuando el atributo del gestor de colas **CONFIGEV** se establece en *ENABLED* y se emite cualquiera de los siguientes mandatos, o su equivalente de MQSC, incluso si no hay ningún cambio real en el registro de autorización:

- Mandato PFC [Suprimir registro de autorización](#)
- Mandato PFC [Establecer registro de autorización](#)
- Mandato de control [setmqaut](#)
- Mandato CL [RVKMQMAUT](#)
- Mandato CL [GRMQMAUT](#)

Cuándo no se generan sucesos de configuración de autorización

Los mensajes de sucesos de configuración de autorización no se generan en las siguientes circunstancias:

- Cuando un mandato falla.
- Cuando un gestor de colas detecta un error al tratar de transferir un mensaje a la cola de sucesos, en cuyo caso el mandato se ejecuta, pero no se genera ningún mensaje de suceso.
- Al crear o suprimir un gestor de colas
- Cuando se suprime un objeto independientemente de la opción **AUTHREC** en el mandato de supresión. El correspondiente suceso de mandato muestra esa operación, que no se aplica al registro de autorización para usuarios individuales.

Conceptos relacionados

“Sucesos de mandatos” en la [página 47](#)

Los sucesos de mandato son notificaciones que un mandato PCF o MQSC ha ejecutado de forma satisfactoria.

Referencia relacionada

[REFRESH QMGR](#)

Programa de ejemplo para supervisar sucesos de instrumentación en Multiplatforms

amqsevt formatea los sucesos de instrumentación que un gestor de colas puede crear y se proporciona con IBM MQ for Multiplatforms. El programa lee mensajes de las colas de sucesos y les da formato de series legibles.

Como programa de ejemplo, se proporcionan tanto el archivo fuente como el archivo binario. El ejemplo se proporciona en todos los Multiplatforms, incluido IBM i.

El archivo binario único `amqsevt` (o `amqsevt.exe`) se entrega en el conjunto de archivos binario y se instala en el directorio `bin` de ejemplos (`tools\c\samples\bin` o `bin64`).

Los archivos fuente `amqsevt.c` también se suministran en el conjunto de archivos de ejemplos y se instala en el directorio de ejemplos, es decir, `tools\c\samples` en Windows.

Tenga en cuenta que el programa puede leer de varias colas de sucesos y suscribirse a varios temas utilizando [MQCB](#) para recuperar los mensajes.

Cuando se ejecuta como cliente, el ejemplo puede conectarse con cualquier gestor de colas incluido z/OS.

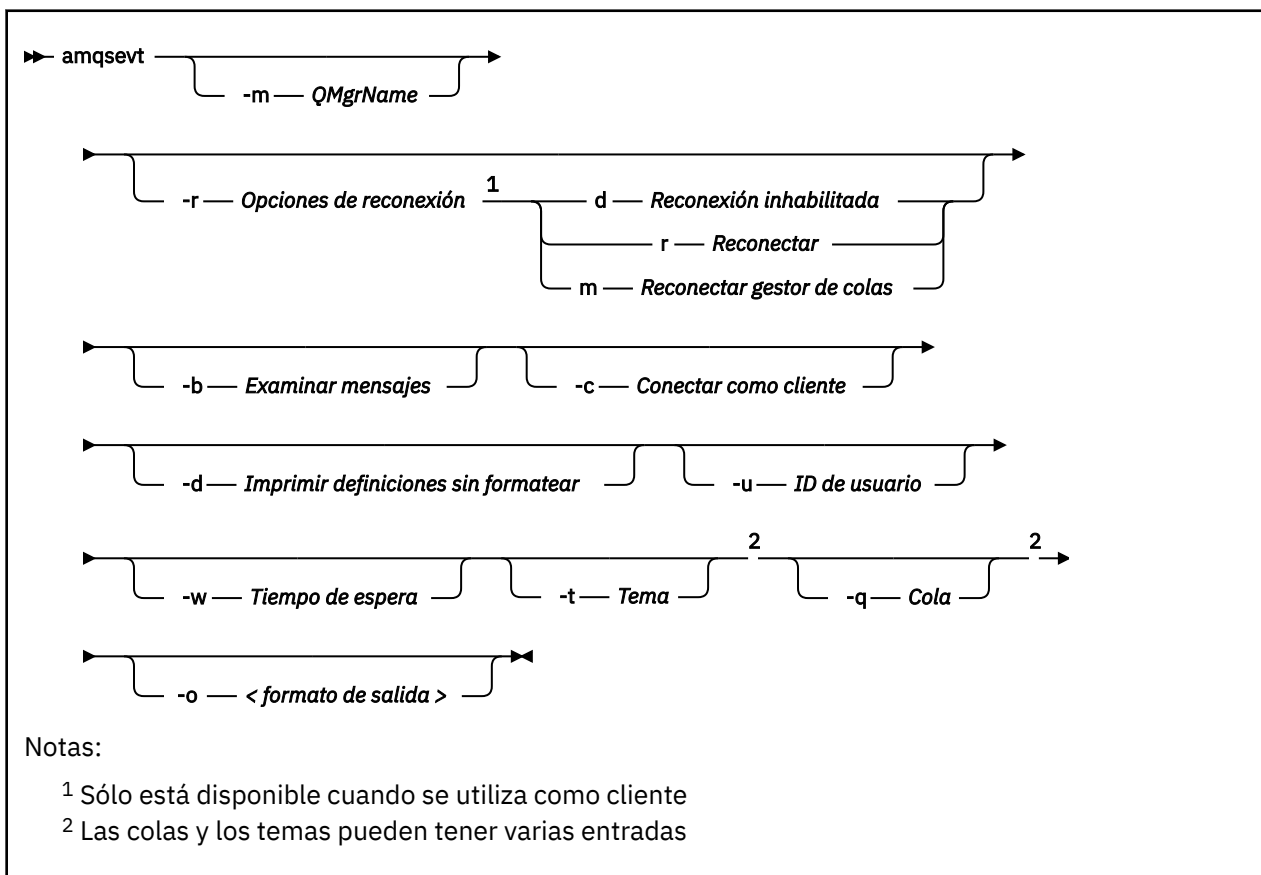


Atención: Puede utilizar el programa sin especificar ningún parámetro, en cuyo caso el programa intentará conectarse al gestor de colas predeterminado y leer mensajes del conjunto estándar de colas de sucesos (`SYSTEM.ADMIN.*.EVENT`).

En esta situación, el programa espera de forma indefinida que lleguen mensajes hasta que se pulsa la tecla Intro para finalizar el programa.

Sin embargo, es más probable utilizar el programa con las distintas opciones descritas.

Sintaxis



Parámetros opcionales

-m QueueManagerName

Especifique un gestor de colas específico para leer sucesos.

-r Reconnection Options

Las opciones de reconexión automática cuando se utilizan como un cliente. Los valores posibles son:

d

Reconexión del cliente inhabilitada

r

Reconectar el cliente

m

Reconectar el gestor de colas

-b

Examinar registros únicamente, en lugar de leer de forma destructiva los mensajes

-c

Selecciona una conexión como un cliente.

-d

Selecciona la modalidad de impresión utilizada en el segundo ejemplo. Las constantes de MQI se imprimen exactamente tal como aparecen en los archivos de cabecera.

-u User ID

Especifique un usuario concreto y obtendrá una solicitud para que especifique la contraseña

-w Wait

Hace que el programa finalice si no ha llegado ningún mensaje de suceso en el número de segundos especificado.

Tenga en cuenta que si no especifica una hora, el programa solo finaliza normalmente cuando pulsa la tecla Intro.

-t Topic y

-q Queue

Las opciones **-q** y **-t** se pueden proporcionar varias veces en la línea de mandatos.

Por lo tanto, es posible leer de algunas colas estándares así como de temas (si los sucesos se envían a ellos) de una sola ejecución del programa.

Si no se indica ningún nombre de cola ni de tema en la línea de mandatos, se abren las colas de sucesos predeterminadas.

Nota: El programa detecta si se ha conectado a un gestor de colas z/OS como cliente y cambia el conjunto predeterminado de colas de sucesos de forma adecuada, ya que z/OS no tiene el SYSTEM.ADMIN.LOGGER.EVENT .

Cuando se utilizan temas, el programa utiliza una suscripción no duradera con una cola gestionada de modo que todo se borre cuando salga.

-o <output format>

El formato de la salida. Los valores posibles son:

text

Formato de texto estándar: este es el valor predeterminado.

json

Formato JSON estándar; cualquier aplicación que comprenda JSON puede tomar esta salida y procesarla directamente.

Salida de ejemplo

Los tres ejemplos siguientes muestran la salida del programa.

El primer ejemplo utiliza la opción de formato predeterminada donde el programa toma la definición MQI de un campo y formatea la salida para que ésta sea más legible.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : Queue Mgr Event
Reason               : Unknown Alias Base Queue
Event created        : 2015/06/17 13:47:07.02 GMT
  Queue Mgr Name     : V8003_A
  Queue Name         : EVT.NO.BASE.QUEUE
  Base Object Name   : EVT.NOT.DEFINED
  Appl Type          : Unix
  Appl Name          : amqsput
  Base Type          : Queue
```

El segundo ejemplo muestra el formato alternativo, utilizando la opción **-d**, que no intenta convertir constantes de la MQI. Esto puede ser preferible para algunas herramientas de script que buscan valores específicos de MQI.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : MQCMD_Q_MGR_EVENT
Reason               : MQRC_UNKNOWN_ALIAS_BASE_Q
Event created        : 2015/06/17 13:52:48.18 GMT
  MQCA_Q_MGR_NAME    : V8003_A
  MQCA_Q_NAME        : EVT.NO.BASE.QUEUE
  MQCA_BASE_OBJECT_NAME : EVT.NOT.DEFINED
  MQIA_APPL_TYPE     : MQAT_UNIX
  MQCACF_APPL_NAME   : amqsput
  MQIA_BASE_TYPE     : MQOT_Q
```

El tercer ejemplo muestra la salida JSON del mandato

```
amqsevt -m QM1 -q SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE -o json
```

```

{
  "eventSource" : { "objectName": "SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE",
                   "objectType" : "Queue" },
  "eventType" : {
    "name" : "Activity Trace",
    "value" : 209
  },
  "eventReason" : {
    "name" : "None",
    "value" : 0
  },
  "eventCreation" : {
    "timeStamp" : "2018-07-10T12:44:26Z",
    "epoch" : 1531226666
  },
  "eventData" : {
    "queueMgrName" : "QM1",
    "hostName" : "<yourhostname>",
    "startDate" : "2018-07-10",
    "startTime" : "13:44:25",
    "endDate" : "2018-07-10",
    "endTime" : "13:44:26",
    "commandLevel" : 910,
    ...
  }
}

```

Ejemplo de uso

El siguiente ejemplo muestra cómo utilizar más de una cola:

```
amqsevt -m QM1 -q SYSTEM.ADMIN.QMGR.EVENT -q SYSTEM.ADMIN.PERM.EVENT -w 1
```

Conceptos relacionados

[“Supervisión de sucesos” en la página 9](#)

La supervisión de sucesos es el proceso de detección de las apariciones de los *sucesos de instrumentación* en una red de gestores de colas. Un suceso de instrumentación es una combinación lógica de sucesos detectada por un gestor de colas o instancia de canal. Dicho suceso hace que el gestor de colas o una instancia de canal coloque un mensaje especial, denominado *mensaje de suceso*, en una cola de sucesos.

[“Sucesos de instrumentación” en la página 10](#)

Un suceso de instrumentación es una combinación lógica de condiciones que un gestor de colas o una instancia de canal detecta y transfiere un mensaje especial, denominado *mensaje de suceso*, a una cola de sucesos.

Referencia relacionada

[Programación C](#)

[“Programa C de ejemplo para supervisar la cola de sucesos del registrador” en la página 51](#)

Utilice esta página para ver un programa de ejemplo, escrito en C, que supervisa la cola de sucesos de registrador en busca de mensajes de suceso nuevos, lee dichos mensajes, y coloca su contenido en la salida estándar (stdout).

Supervisión de mensajes

La supervisión de mensajes es el proceso de identificar la ruta que ha seguido un mensaje a través de una red de gestores de colas. Al identificar los tipos de actividades, y la secuencia de actividades realizadas en nombre de un mensaje, es posible determinar la ruta del mensaje.

Cuando un mensaje pasa a través de una red de gestores de colas, varios procesos realizan actividades en nombre del mensaje. Utilice una de las técnicas siguientes para determinar la ruta de un mensaje:

- La aplicación de visualización de ruta de IBM MQ (dspmqrte)
- Registro de actividad
- Mensajería de ruta de rastreo

Estas técnicas generan mensajes especiales que contienen información sobre todas las actividades realizadas en el mensaje a medida que pasa a través de una red de gestores de colas. Utilice la información que se devuelve en esos mensajes especiales para alcanzar los siguientes objetivos:

- Registrar la actividad de mensajes.
- Determinar la última ubicación conocida de un mensaje.
- Detectar los problemas de direccionamiento en la red de gestores de colas.
- Ayudar a determinar las causas de los problemas de direccionamiento de la red de gestores de colas.
- Comprobar que la red del gestor de colas funciona correctamente.
- Familiarizarse con la ejecución de la red del gestor de colas.
- Mensajes de rastreo publicados.

Conceptos relacionados

[Tipos de mensajes](#)

Actividades y operaciones

Las actividades son acciones independientes que una aplicación realiza en nombre de un mensaje. Las actividades constan de operaciones, que son tareas individuales que una aplicación lleva a cabo.

Las acciones siguientes son ejemplos de actividades:

- Un agente de canal de mensajes (MCA) envía un mensaje de una cola de transmisión a través de un canal.
- Un MCA recibe un mensaje de un canal y lo coloca en su cola de destino.
- Una aplicación que obtiene un mensaje de una cola, y que coloca un mensaje de respuesta, como respuesta.
- El motor de publicación/suscripción de IBM MQ procesa un mensaje.

Las actividades constan de una o varias *operaciones*. Las operaciones son tareas individuales que una aplicación lleva a cabo. Por ejemplo, la actividad de un MCA que envía un mensaje desde una cola de transmisión a través de un canal consta de las operaciones siguientes:

1. Obtener un mensaje de una cola de transmisión (una operación *Get*).
2. Enviar el mensaje a través de un canal (una operación *Send*).

En una red de publicación/suscripción, la actividad del motor de publicación/suscripción de IBM MQ que procesa un mensaje puede constar de las siguientes operaciones:

1. Colocar un mensaje en una serie de tema (una operación *Put*).
2. Cero o más operaciones para cada uno de los suscriptores que se tienen en cuenta para la recepción del mensaje (una operación *Publish*, una operación *Discarded Publish*, o una operación *Excluded Publish*).

Información de las actividades

Puede identificar la secuencia de las actividades realizadas en un mensaje registrando información mientras se direcciona a través de una red de gestores de colas. Puede determinar la ruta de un mensaje a través de la red de gestores de colas a partir de la secuencia de actividades realizadas en el mensaje, y puede obtener la información siguiente:

La última ubicación conocida de un mensaje

Si un mensaje no llega a su destino previsto, puede determinar la última ubicación conocida del mensaje a partir una ruta de mensaje completa o parcial.

Problemas de configuración surgidos en una red de gestores de colas

Al estudiar la ruta de un mensaje a través de una red de gestores de colas, es posible que vea que el mensaje no se ha llegado a donde se esperaba. Hay muchos motivos por los que esto puede ocurrir, por ejemplo, si un canal está inactivo, el mensaje puede tomar una ruta alternativa.

Para una aplicación de publicación/suscripción, también puede determinar la ruta de un mensaje que se publica en un tema y todos los mensajes que fluyen dentro de una red de gestores de colas como resultado de que se publiquen a los suscriptores.

En tales situaciones, un administrador del sistema puede determinar si existe algún problema en la red de gestores de colas, y si resulta pertinente, puede corregirlos.

Rutas de mensajes

En función del motivo por el cual determine una ruta de mensaje, puede utilizar los enfoques generales siguientes:

Utilizando la información de actividad registrada para un mensaje de ruta de rastreo

Los mensajes de ruta de rastreo registran la información de actividad para una finalidad específica. Puede utilizarlos para determinar los problemas de configuración que hayan surgido en una red de gestores de colas, o para determinar la última ubicación conocida de un mensaje. Si se genera un mensaje de ruta de rastreo para determinar la última ubicación conocida de un mensaje que no llega a su destino previsto, puede imitar al mensaje original. Esto proporciona al mensaje de ruta de rastreo la mejor posibilidad de tomar la ruta que haya seguido el mensaje original.

La aplicación de visualización de ruta de IBM MQ puede generar mensajes de ruta de rastreo.

Utilizando la información de actividad registrada para el mensaje original

Puede habilitar cualquier mensaje para el registro de actividad y hacer que la información de actividad se registre en su nombre. Si un mensaje no llega a su destino previsto, puede utilizar la información de actividad registrada para determinar la última ubicación conocida del mensaje. Al utilizar la información de actividad del mensaje original, se puede determinar la ruta de mensaje más precisa posible, que le llevará a la última ubicación conocida. Para utilizar este enfoque, se debe haber habilitado el registro de actividad en el mensaje original.

Aviso: Evite habilitar el registro de actividad de todos los mensajes de una red de gestores de colas. Los mensajes habilitados para el registro de actividad pueden tener muchos informes de actividad que se hayan generado en su nombre. Si se habilita registro de actividad de todos los mensajes de una red de gestores de colas, el tráfico de red del gestor de colas puede aumentar hasta alcanzar un nivel inaceptable.

Conceptos relacionados

[“Supervisión de mensajes” en la página 60](#)

La supervisión de mensajes es el proceso de identificar la ruta que ha seguido un mensaje a través de una red de gestores de colas. Al identificar los tipos de actividades, y la secuencia de actividades realizadas en nombre de un mensaje, es posible determinar la ruta del mensaje.

[“Técnicas de ruta de mensajes” en la página 62](#)

El registro de actividad y la mensajería de ruta de rastreo son técnicas que le permiten registrar información de actividad para un mensaje a medida que éste se direcciona a través de una red de gestores de colas.

[“Mensajería de ruta de rastreo” en la página 69](#)

La mensajería de ruta de rastreo es una técnica que utiliza *mensajes de ruta de rastreo* para registrar información de actividad para un mensaje. La mensajería de ruta de rastreo implica el envío de un mensaje de ruta de rastreo a una red de gestores de colas.

Tareas relacionadas

[Escribir sus propios agentes de canales de mensajes](#)

Técnicas de ruta de mensajes

El registro de actividad y la mensajería de ruta de rastreo son técnicas que le permiten registrar información de actividad para un mensaje a medida que éste se direcciona a través de una red de gestores de colas.

Registro de actividad

Si en un mensaje se ha especificado la opción de informe pertinente, aquél solicita que las aplicaciones generen *informes de actividad* mientras se direcciona a través de una red de gestores de colas. Cuando una aplicación realiza una actividad en nombre de un mensaje, se puede generar y entregar un informe de actividad en una ubicación que resulte apropiada. Un informe de actividad contiene información sobre la actividad que se ha efectuado en el mensaje.

La información de actividad recopilada mediante informes de actividad debe colocarse en orden antes de que se pueda determinar una ruta de mensaje.

Mensajería de ruta de rastreo

La *mensajería de ruta de rastreo* es una técnica que implica el envío de un *mensaje de ruta de rastreo* a una red de gestores de colas. Cuando una aplicación realiza una actividad en nombre del mensaje de ruta de rastreo, la información de actividad se pueden acumular en los datos de mensaje del mensaje de ruta de rastreo, o se pueden generar informes de actividad. Si la información de actividad se acumula en los datos de mensaje del mensaje de ruta de rastreo, cuando llega a su cola de destino, se puede generar un mensaje de respuesta de ruta de rastreo que contiene toda la información del mensaje de ruta de rastreo, que puede entregarse en una ubicación adecuada.

Puesto que un mensaje de ruta de rastreo se dedica a registrar la secuencia de actividades realizadas en su nombre, hay más opciones de proceso disponibles en comparación con los mensajes normales que solicitan informes de actividad.

Comparación del registro de actividad y la mensajería de ruta de rastreo

Tanto el registro de la actividad como la mensajería de ruta de rastreo pueden proporcionar información de la actividad para poder determinar la ruta que un mensaje ha tomado a través de una red de gestores de colas. Ambos métodos tienen sus propias ventajas.

Ventaja	Registro de actividad	Mensajería de ruta de rastreo
Puede determinar la última ubicación conocida de un mensaje	Sí	Sí
Puede determinar los problemas de configuración surgidos en una red de gestores de colas	Sí	Sí
La puede solicitar cualquier mensaje (no está restringida al uso con mensajes de ruta de rastreo)	Sí	No
Los datos del mensaje no se modifican	Sí	No
El mensaje se procesa normalmente	Sí	No
La información de actividad puede acumularse en los datos del mensaje	No	Sí
Entrega de mensajes opcional en la cola de destino	No	Sí
Si un mensaje queda atrapado en un bucle infinito, se puede detectar y tratar de resolver el problema	No	Sí
La información de actividad se puede poner en orden de forma fiable	No	Sí
Se proporciona una aplicación para visualizar la información de actividad	No	Sí

Finalización de la ruta de mensaje

En algunos casos, no es posible identificar la secuencia completa de las actividades realizadas en nombre de un mensaje, de modo que sólo se puede determinar una ruta de mensaje parcial. La finalización de una

ruta de mensaje depende directamente de la red de gestores de colas a través de la cual se direccionan los mensajes.

Los MCA y las aplicaciones escritas por el usuario conectadas a gestores de colas pueden registrar información relacionada con las actividades realizadas en nombre de un mensaje. Los atributos de gestor de colas ACTIVREC y ROUTEREC controlan el registro de la información de actividad. Una red de gestores de colas puede determinar rutas de mensajes completas.

Cómo se almacena la información de actividad

IBM MQ almacena la información de actividad en informes de actividad, mensajes de ruta de rastreo o mensajes de respuesta de ruta de rastreo. En cada caso, la información se almacena en una estructura que se conoce como el grupo PCF *Activity*. Un mensaje de ruta de rastreo o mensaje de respuesta de ruta de rastreo puede contener muchos grupos PCF *Activity*, dependiendo del número de actividades realizadas en el mensaje. Los informes de actividad contienen un grupo PCF *Activity* porque se genera un informe de actividad distinto para cada actividad registrada.

Con la mensajería de ruta de rastreo se puede registrar información adicional. Esta información adicional se almacena en una estructura que se conoce como el grupo PCF *TraceRoute*. El grupo PCF *TraceRoute* contiene varias estructuras PCF que se utilizan para almacenar información adicional de la actividad, y para especificar las opciones que determinan cómo se maneja el mensaje de ruta de rastreo mientras se direcciona a través de una red de gestores de colas.

Conceptos relacionados

[“Registro de actividad” en la página 64](#)

El registro de la actividad es una técnica para determinar las rutas que los mensajes toman en una red de gestores de colas. Para determinar la ruta que ha seguido un mensaje, se registran las actividades realizadas en nombre del mensaje.

[“Mensajería de ruta de rastreo” en la página 69](#)

La mensajería de ruta de rastreo es una técnica que utiliza *mensajes de ruta de rastreo* para registrar información de actividad para un mensaje. La mensajería de ruta de rastreo implica el envío de un mensaje de ruta de rastreo a una red de gestores de colas.

Referencia relacionada

[“Grupo PCF *TraceRoute*” en la página 75](#)

Los atributos del grupo PCF *TraceRoute* controlan el comportamiento de un mensaje de ruta de rastreo. El grupo *TraceRoute* está en los datos de mensaje de cada mensaje de ruta de rastreo.

[“Datos de mensaje de informe de actividad” en la página 110](#)

Utilice esta página para ver los parámetros el grupo PCF *Activity* contiene en un mensaje de informe de actividad. Algunos parámetros sólo se devuelven cuando se han realizado operaciones específicas.

Registro de actividad

El registro de la actividad es una técnica para determinar las rutas que los mensajes toman en una red de gestores de colas. Para determinar la ruta que ha seguido un mensaje, se registran las actividades realizadas en nombre del mensaje.

Cuando se utiliza el registro de actividad, cada actividad realizada en nombre de un mensaje puede registrarse en un informe de actividad. Un informe de actividad es un tipo de mensaje de informe. Cada informe de actividad contiene información sobre la aplicación que ha realizado la actividad en nombre del mensaje, cuándo tuvo lugar la actividad e información sobre las operaciones que se han realizado como parte de la actividad. Los informes de actividad se entregan normalmente a una cola de respuesta en la que se recopilan. Estudiando la actividad de los informes relacionados con un mensaje, puede determinar la ruta que tomó un mensaje a través de la red del gestor de colas.

Uso del informe de actividad

Cuando los mensajes se direccionan a través de una red de gestores de colas, se pueden generar los informes de actividad. Puede utilizar la información del informe de actividad de las siguientes maneras:

Determinar la última ubicación conocida de un mensaje

Si un mensaje que está habilitado para el registro de la actividad no llega a su destino previsto, es posible estudiar los informes de actividad generados para el mensaje según se ha direccionado a través de una red de gestores de colas para determinar la última ubicación conocida del mensaje.

Determinar los problemas de configuración con una red de gestores de colas

Pueden enviarse varios mensajes habilitados para el registro de la actividad a una red de gestores de colas. Estudiando los informes de actividad relacionados con cada mensaje puede ser evidente que no han tomado la ruta prevista. Hay muchos motivos por los que esto puede producirse, por ejemplo, un canal podría haberse detenido, forzando al mensaje a tomar una ruta alternativa. En estas situaciones, un administrador del sistema puede determinar si hay algún problema en la red de gestores de colas y, en caso afirmativo, corregirlo.

Nota: Puede utilizar el registro de actividad junto con los mensajes de ruta de rastreo mediante la aplicación de visualización de ruta de IBM MQ.

Formato de informe de actividad

Los informes de actividad son mensajes PCF generados por las aplicaciones que han realizado una actividad en nombre de un mensaje. Los informes de actividad son mensajes de informe estándar de IBM MQ que contienen un descriptor de mensaje y datos de mensaje, como se indica a continuación:

El descriptor del mensaje

- Una estructura MQMD

Datos de mensaje

- Una cabecera PCF incluida (MQEPH)
- Datos de mensaje de informe de actividad

Los datos de mensaje de informe de actividad incluyen el grupo PCF *Activity* y, si se han generado para un mensaje de ruta de rastreo, el grupo PCF *TraceRoute*.

Referencia relacionada

[MQMD - Descriptor de mensaje](#)

[MQEPH - Cabecera PCF incrustada](#)

Control del registro de actividad

Habilitar el registro de actividad en el nivel de gestor de colas. Para habilitar toda una red de gestores, habilite individualmente cada gestor de colas de la red para el registro de actividad. Si habilita más gestores de colas, se generan más informes de actividad.

Acerca de esta tarea

Para generar informes de actividad para un mensaje a medida que se direcciona a través de un gestor de colas: defina el mensaje para que solicite los informes de actividad; habilite el gestor de colas para el registro de actividad; y asegúrese de que las aplicaciones que realizan las actividades del mensaje sean capaces de generar informes de actividad.

Si no desea que se generen informes de actividad para un mensaje a medida que se direccionan a través de un gestor de colas, *inhabilite* el gestor de colas para el registro de actividades.

Procedimiento

1. solicitar informes de actividad para un mensaje
 - a) En el descriptor de mensaje del mensaje, especifique MQRO_ACTIVITY en el campo *Report*.
 - b) En el descriptor de mensaje del mensaje, especifique el nombre de una cola de respuesta en el campo *ReplyToQ*.

Aviso: Evite habilitar el registro de actividad de todos los mensajes de una red de gestores de colas. Los mensajes habilitados para el registro de actividad pueden tener muchos informes de actividad que se hayan generado en su nombre. Si se habilita registro de actividad de todos los mensajes de una red de gestores de colas, el tráfico de red del gestor de colas puede aumentar hasta alcanzar un nivel inaceptable.

2. Habilitar o inhabilitar el registro de actividad en el gestor de colas.

Utilice el mandato MQSC ALTER QMGR, especificando el parámetro ACTIVREC, para cambiar el valor del atributo de gestor de colas. El valor puede ser:

MSG

El gestor de colas está habilitado para el registro de actividad. Todos los informes de actividad generados se entregan en la cola de respuesta especificada en el descriptor de mensaje del mensaje. Éste es el valor predeterminado.

COLA

El gestor de colas está habilitado para el registro de actividad. Todos los informes de actividad generados se entregan en la cola del sistema local SYSTEM.ADMIN.ACTIVITY.QUEUE. La cola del sistema también se puede utilizar para reenviar los informes de actividad a una cola común.

DISABLED

El gestor de colas está inhabilitado para el registro de actividad. No se genera ningún informe de actividad mientras se esté en el ámbito de este gestor de colas.

Por ejemplo, para habilitar un gestor de colas para el registro de actividad, y especificar que todos los informes de actividad generados se entreguen en la cola del sistema local SYSTEM.ADMIN.ACTIVITY.QUEUE, utilice el mandato MQSC siguiente:

```
ALTER QMGR ACTIVREC(Queue)
```

Recuerde: Cuando modifique el atributo de gestor de colas *ACTIVREC*, un MCA que esté en ejecución no detectará el cambio efectuado hasta que se reinicie el canal.

3. Asegúrese de que la aplicación utilice el mismo algoritmo que utilizan los MCA para determinar si se debe generar un informe de actividad para un mensaje:
 - a) Verifique que el mensaje haya solicitado que se generen los informes de actividad.
 - b) Verifique que el gestor de colas en el que reside el mensaje actualmente esté habilitado para el registro de actividad.
 - c) Transfiera el informe de actividad a la cola que determine el atributo de gestor de colas *ACTIVREC*.

Configuración de una cola común para informes de actividad

Para determinar las ubicaciones de los informes de actividad relacionados con un mensaje específico cuando se entregan los informes a la cola del sistema local, resulta más eficaz utilizar una cola común en un único nodo.

Antes de empezar

Establezca el parámetro **ACTIVREC** para habilitar el registro de actividad en el gestor de colas y especificar que los informes de actividad generados se entreguen en la cola del sistema local SYSTEM.ADMIN.ACTIVITY.QUEUE.

Acerca de esta tarea

Si se establece un número de gestores de colas en una red de gestores de colas para entregar los informes de actividad en la cola del sistema local, se puede tardar mucho tiempo en determinar las ubicaciones de los informes de actividad relacionados con un mensaje específico. De forma alternativa, utilice un único nodo, que es un gestor de colas en el que se aloja una cola común. Todos los gestores de colas de una red de gestores de colas pueden entregar los informes de actividad en esta cola común. La ventaja de utilizar una cola común es que los gestores de colas no tienen que entregar los informes de actividad en la cola de respuesta especificada en un mensaje y, al determinar las ubicaciones de los informes de actividad relacionados con un mensaje, sólo se consulta una cola.

Para configurar una cola común, efectúe los pasos siguientes:

Procedimiento

1. Seleccione o defina un gestor de colas como el único nodo.
2. En el único nodo, seleccione o defina una cola para utilizarla como la cola común
3. En todos los gestores de colas en los que los informes de actividad se vayan a entregar en la cola común, vuelva a definir la cola del sistema local SYSTEM.ADMIN.ACTIVITY.QUEUE como una definición de cola remota:
 - a) Especifique el nombre del nodo único como el nombre del gestor de colas remoto.
 - b) Especifique el nombre de la cola común como el nombre de la cola remota.

Determinación de la información de ruta del mensaje

Para determinar la ruta de un mensaje, obtenga la información de los informes de actividad recopilados. Determine si hay suficientes informes de actividad en la cola de respuesta para permitirle determinar la información necesaria, y colocar los informes de actividad por orden.

Acerca de esta tarea

El orden en que los informes de actividad se transfieren a la cola de respuesta no se correlaciona necesariamente con el orden en que se han realizado las actividades. Debe ordenar los informes de actividad manualmente, a menos que se generen para un mensaje de ruta de rastreo, en cuyo caso puede utilizar la aplicación de visualización de ruta de IBM MQ para ordenarlos.

Determine si hay suficientes informes de actividad en la cola de respuesta para poder obtener la información necesaria:

Procedimiento

1. Identifique todos los informes de actividad relacionados en la cola de respuesta, comparando los identificadores de los informes de actividad y el mensaje original. Asegúrese de establecer la opción de informe del mensaje original, de modo que los informes de actividad se pueden correlacionar con el mensaje original.
2. Ordene los informes de actividad identificados de la cola de respuesta.
Puede utilizar los parámetros siguientes del informe de actividad:

OperationType

Los tipos de operaciones realizadas pueden permitirle determinar el informe de actividad que se haya generado directamente antes, o después, del informe de actividad actual.

Por ejemplo, en un informe de actividad se detalla que un MCA ha enviado un mensaje de una cola de transmisión a través de un canal. La última operación que se ha detallado en el informe de actividad tiene un valor de *OperationType* de *send* y se detalla que el mensaje se ha enviado utilizando el canal, CH1, al gestor de colas de destino, QM1. Esto significa que la actividad siguiente realizada en el mensaje se habrá producido en el gestor de colas, QM1, y que habrá empezado con una operación de tipo *receive* procedente del canal, CH1. Mediante esta información puede identificar el informe de actividad siguiente, siempre que éste exista y se haya obtenido.

OperationDate y OperationTime

Puede determinar el orden general de las actividades a partir de las fechas y horas de las operaciones de cada informe de actividad.

Aviso: A menos que cada gestor de colas de la red de gestores de colas tenga los relojes del sistema sincronizados, ordenar por fecha y hora no garantiza que los informes de actividad aparezcan en el orden correcto. Debe establecer el orden manualmente.

El orden de los informes actividad representa la ruta, o la ruta parcial, que ha tomado el mensaje a través de la red de gestores de colas.

3. Obtenga la información que necesite de la información de actividad, de los informes de actividad que ha ordenado.

Si tiene información suficiente sobre el mensaje, es posible que pueda obtener más informes de actividad.

Recuperación de más registros de actividad

Para determinar una ruta de mensaje, debe haber disponible información suficiente en los informes de actividad recopilados. Si recupera los informes de actividad relacionados con un mensaje de la cola de respuesta que el mensaje ha especificado, pero no tiene la información necesaria, busque otros informes de actividad.

Acerca de esta tarea

Para determinar las ubicaciones de otros informes de actividad, efectúe los pasos siguientes:

Procedimiento

1. Para los gestores de colas de la red de gestores de colas que entreguen informes de actividad en una cola común, recupere los informes de actividad de la cola común que tenga un valor *CorrelId* que coincida con el valor *MsgId* del mensaje original.
2. Para los gestores de colas de la red de gestores de colas que no entreguen informes de actividad en una cola común, recupere los informes de actividad tal como se indica a continuación:
 - a) Examine los informes de actividad existentes para identificar los gestores de colas a través de los cuales se direcciona el mensaje.
 - b) Para dichos gestores de colas, identifique los gestores de colas que estén habilitados para el registro de la actividad.
 - c) Para dichos gestores de colas, identifique aquellos que no hayan devuelto los informes de actividad a la cola de respuesta especificada.
 - d) Para cada uno de los gestores de colas que identifique, compruebe la cola del sistema SYSTEM.ADMIN.ACTIVITY.QUEUE, y recupere los informes de actividad que tengan un valor *CorrelId* que coincida con el valor *MsgId* del mensaje original.
 - e) Si no encuentra ningún informe de actividad en la cola del sistema, compruebe la cola de mensajes no entregados del gestor de colas, si existe ésta.
Un informe de actividad sólo puede entregarse a una cola de mensajes no entregados si se establece la opción de informe MQRO_DEAD_LETTER_Q.

3. Ordene todos los informes de actividad.

El orden de los informes actividad representa la ruta, o la ruta parcial, que ha tomado el mensaje.

4. Obtenga la información que necesite de la información de actividad, de los informes de actividad que ha ordenado.

En algunas circunstancias, la información de actividad registrada no puede alcanzar la cola de respuesta especificada, una cola común, o una cola del sistema.

Circunstancias en las que no se adquiere la información de actividad

Para poder determinar la secuencia completa de las actividades llevadas a cabo en nombre de un mensaje, debe obtenerse la información relativa a cada actividad. Si no se ha registrado, o no se ha obtenido, la información relacionada con una actividad cualquiera, sólo puede determinar sólo secuencia parcial de la misma.

La información de actividad no se registra en las circunstancias siguientes:

- El mensaje lo procesa un gestor de colas anterior a IBM WebSphere MQ 6.0.
- El mensaje lo procesa un gestor de colas que no está habilitado para el registro de actividad.
- No se está ejecutando la aplicación que se esperaba que procesara el mensaje.

La información de actividad registrada no puede llegar a la cola de respuesta en las especificada en las circunstancias siguientes:

- No hay ningún canal definido para direccionar los informes de actividad a la cola de respuesta.
- No se está ejecutando el canal a través del que direccionar los informes de actividad a la cola de respuesta.
- No se ha especificado la definición de cola remota a través de la cual se deben devolver los informes de actividad al gestor de colas donde reside la cola de respuesta (el alias de gestor de colas).
- El usuario que ha generado el mensaje original no tiene autorización suficiente para las operaciones de abrir, o transferir, respecto al alias de gestor de colas.
- El usuario que ha generado el mensaje original no tiene autorización suficiente para las operaciones de abrir, o transferir, respecto a la cola de respuesta.
- La cola de respuesta tiene la transferencia inhibida.

La información de actividad registrada no puede llegar a la cola del sistema, o a una cola común, en las circunstancias siguientes:

- Si se debe utilizar una cola común y no se ha definido ningún canal para direccionar los informes de actividad a la cola común.
- Si se debe utilizar una cola común y no se está ejecutando el canal a través del cual se deben direccionar los informes de actividad a la cola común.
- Si se debe utilizar una cola común y la cola del sistema se ha definido de forma incorrecta.
- El usuario que ha generado el mensaje original no tiene autorización suficiente para las operaciones de abrir, o transferir, respecto a la cola de sistema.
- La cola de sistema tiene la transferencia inhibida.
- Si se debe utilizar una cola común y el usuario que ha generado el mensaje original no tiene autorización suficiente para las operaciones de abrir, o transferir, respecto a la cola común.
- Si se debe utilizar una cola común y ésta tiene la transferencia inhibida.

En estas circunstancias, siempre que el informe de actividad no tenga especificada la opción MQRO_DISCARD_MSG, el informe de actividad se puede recuperar de una cola de mensajes no entregados, si se ha definido una en el gestor de colas en el que se ha rechazado el informe de actividad. Un informe de actividad sólo tendrá especificada esta opción de informe si el mensaje original, a partir del cual se ha generado el informe de actividad, tenía especificado MQRO_PASS_DISCARD_AND_EXPIRY y MQRO_DISCARD_MSG en el campo Report del descriptor de mensaje.

Mensajería de ruta de rastreo

La mensajería de ruta de rastreo es una técnica que utiliza *mensajes de ruta de rastreo* para registrar información de actividad para un mensaje. La mensajería de ruta de rastreo implica el envío de un mensaje de ruta de rastreo a una red de gestores de colas.

A medida que el mensaje de ruta de rastreo se direcciona a través de la red de gestores de colas, se registra información sobre su actividad. Esta información de actividad incluye información sobre las aplicaciones las actividades han realizado, cuándo se han realizado, y las operaciones que se han realizado como parte de las actividades. Puede utilizar la información registrada mediante la mensajería de ruta de rastreo, para los fines siguientes:

Determinar la última ubicación conocida de un mensaje.

Si un mensaje no llega a su destino previsto, puede utilizar la información de actividad registrada para un mensaje de ruta de rastreo para determinar la última ubicación conocida del mensaje. Un mensaje de ruta de rastreo se envía a una red de gestores de colas que tengan el mismo destino objetivo que el mensaje original, con la intención de que siga la misma ruta. La información de actividad se puede acumular en los datos de mensaje del mensaje de ruta de rastreo, o se puede registrar mediante los informes de actividad. Para aumentar la probabilidad de que el mensaje de ruta de rastreo siga la misma ruta que el mensaje original, puede modificar el mensaje de ruta de rastreo para que imite al mensaje original.

Determinar los problemas de configuración surgidos en una red de gestores de colas.

Los mensajes de ruta de rastreo se envían a una red de gestores de colas y se registra la información de actividad. Al estudiar la información de actividad grabada para un mensaje de ruta de rastreo, puede quedar claro que el mensaje de ruta de rastreo no ha seguido la ruta prevista. Hay muchos motivos por los que esto puede ocurrir, por ejemplo, un canal puede estar inactiva, obligando al mensaje a tomar una ruta alternativa. En estas situaciones, un administrador del sistema puede determinar si hay algún problema en la red de gestores de colas y, en caso afirmativo, corregirlo.

Puede utilizar la aplicación de visualización de ruta de IBM MQ para configurar, generar y colocar mensajes de ruta de rastreo en una red de gestores de colas.

Aviso: Si coloca un mensaje de ruta de rastreo en una lista de distribución, los resultados son imprevisibles.

Conceptos relacionados

[“Referencia de mensaje de ruta de rastreo” en la página 127](#)

Utilice esta página para obtener una visión general del formato de mensaje de ruta de rastreo. Los datos del mensaje de ruta de rastreo incluyen parámetros que describen las actividades que el mensaje de ruta de rastreo ha provocado.

Cómo se registra la información de actividad

Con la mensajería de ruta de rastreo, puede registrar la información de actividad en los datos de mensaje del mensaje de ruta de rastreo, o puede utilizar los informes de actividad. De forma alternativa, puede utilizar ambas técnicas.

Acumulación de la información de actividad en los datos de mensaje del mensaje de ruta de rastreo

A medida que un mensaje de ruta de rastreo se direcciona a través de una red de gestores de colas la información sobre las actividades realizadas en nombre del mensaje de ruta de rastreo se pueden acumular en los datos de mensaje del mensaje de ruta de rastreo. La información de actividad se almacena en grupos PCF *Activity*. Para cada actividad realizada en nombre del mensaje de ruta de rastreo, se graba un grupo PCF *Activity* al final del bloque PCF de los datos del mensaje de ruta de rastreo.

La información de actividad adicional se registra en la mensajería de ruta de rastreo, en un grupo PCF llamado *TraceRoute*. La información de actividad adicional se almacena en este grupo PCF, y se puede utilizar para ayudar a determinar la secuencia de las actividades registradas. Esta técnica se controla mediante el parámetro *Accumulate* en el grupo PCF *TraceRoute*.

Registro de la información de actividad mediante los informes de actividad

A medida que un mensaje de ruta de rastreo se direcciona a través de una red de gestores de colas, se puede generar un informe de actividad para cada actividad que se haya llevado a cabo en nombre del mensaje de ruta de rastreo. La información de actividad se almacena en el grupo PCF *Activity*. Para cada actividad realizada en nombre de un mensaje de ruta de rastreo, se genera un informe de actividad que contiene un grupo PCF *Activity*. El registro de actividad para los mensajes de ruta de rastreo funciona del mismo modo que para cualquier otro mensaje.

Los informes de actividad que se generan para los mensajes de ruta de rastreo contienen información de actividad adicional, en comparación con los generados para cualquier otro mensaje. La información adicional se devuelve en un grupo PCF *TraceRoute*. La información contenida en el grupo PCF *TraceRoute* sólo es precisa desde el momento en que genera el informe de actividad. Puede utilizar la información adicional para que le ayuda a determinar la secuencia de las actividades realizadas en nombre del mensaje de ruta de rastreo.

Adquirir información de actividad registrada

Cuando un mensaje de ruta de rastreo ha llegado a su destino previsto, o se descarta, el método que se utiliza para obtener la información de actividad depende de la forma en que se haya registrado la misma.

Antes de empezar

Si no está familiarizado con la actividad de la información, consulte [“Cómo se registra la información de actividad”](#) en la página 70.

Acerca de esta tarea

Utilice los métodos siguientes para obtener la información de actividad después de que el mensaje de ruta de rastreo haya llegado a su destino previsto, o se descarte:

Procedimiento

- Recupere el mensaje de ruta de rastreo.

El parámetro *Deliver*, del grupo *PCFTraceRoute*, controla si se coloca un mensaje de ruta de rastreo en la cola de destino cuando llega, o si se descarta. Si el mensaje de ruta de rastreo se entrega en la cola de destino, puede recuperarlo de dicha cola. A continuación, puede utilizar la aplicación de visualización de ruta de IBM MQ para visualizar la información de actividad.

Para solicitar que la información de actividad se acumule en los datos de mensaje de un mensaje de ruta de rastreo, establezca el parámetro *Accumulate* del grupo *PCF TraceRoute* en `MQROUTE_ACCUMULATE_IN_MSG`.

- Utilice un mensaje de respuesta de ruta de rastreo.

Cuando un mensaje de ruta de rastreo llega a su destino previsto, o no se puede direccionar más dentro de una red de gestores de colas, se puede generar un mensaje de respuesta de ruta de rastreo. Un mensaje de respuesta de ruta de rastreo contiene un duplicado de toda la información de actividad del mensaje de ruta de rastreo, y se puede entregar en una cola de respuesta que se haya especificado, o en la cola del sistema `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`. Puede utilizar la aplicación de visualización de ruta de IBM MQ para visualizar la información de actividad.

Para solicitar un mensaje de respuesta de ruta de rastreo, establezca el parámetro *Accumulate* del grupo *PCF TraceRoute* en `MQROUTE_ACCUMULATE_AND_REPLY`.

- Utilice los informes de actividad.

Si se generan informes de actividad para un mensaje de ruta de rastreo, deberá localizar los informes de actividad para poder adquirir la información de actividad. A continuación, para determinar la secuencia de las actividades, deberá ordenar los informes de actividad.

Control de mensajería de ruta de rastreo

Habilite la mensajería de ruta de rastreo a nivel de gestor de colas, de modo que las aplicaciones que se encuentren en el ámbito de dicho gestor de colas puedan grabar información de actividad en un mensaje de ruta de rastreo. Para habilitar toda una red de gestores, habilite individualmente cada gestor de colas de la red para la mensajería de ruta de rastreo. Si habilita más gestores de colas, se generan más informes de actividad.

Antes de empezar

Si utiliza los informes de actividad para registrar la información de actividad de un mensaje de ruta de rastreo, consulte [“Control del registro de actividad”](#) en la página 65.

Acerca de esta tarea

Para registrar información de actividad de un mensaje de ruta de rastreo mientras se direcciona a través de una red de gestores de colas, efectúe los pasos siguientes:

Procedimiento

- Defina cómo se debe registrar la información de actividad del mensaje de ruta de rastreo. Consulte [“Generación y configuración de un mensaje de ruta de rastreo”](#) en la página 74.
- Si desea acumular información de actividad en el mensaje de ruta de rastreo, asegúrese de que el gestor de colas esté habilitado para la mensajería de ruta de rastreo.

- Si desea acumular información de actividad en el mensaje de ruta de rastreo, asegúrese de que las aplicaciones que realicen actividades en el mensaje de ruta de rastreo sean capaces de grabar la información de actividad en los datos de mensaje del mensaje de ruta de rastreo.

Conceptos relacionados

“Generación y configuración de un mensaje de ruta de rastreo” en la página 74

Un mensaje de ruta de rastreo consta del descriptor de mensaje específico y de los componentes de datos de mensaje. Para generar un mensaje de ruta de rastreo, cree el mensaje manualmente o utilice la aplicación de visualización de ruta de IBM MQ.

Tareas relacionadas

“Control del registro de actividad” en la página 65

Habilitar el registro de actividad en el nivel de gestor de colas. Para habilitar toda una red de gestores, habilite individualmente cada gestor de colas de la red para el registro de actividad. Si habilita más gestores de colas, se generan más informes de actividad.

Habilitación de gestores de colas para la mensajería de ruta de rastreo

Para controlar si los gestores de colas están habilitados o inhabilitados para la mensajería de ruta de rastreo, utilice el atributo de gestor de colas ROUTEREC.

Utilice el mandato MQSC ALTER QMGR, especificando el parámetro ROUTEREC, para cambiar el valor del atributo de gestor de colas. El valor puede ser cualquiera de los valores siguientes:

MSG

El gestor de colas está habilitado para la mensajería de ruta de rastreo. Las aplicaciones dentro del ámbito del gestor de colas pueden grabar información de actividad en el mensaje de ruta de rastreo.

Si el parámetro *Accumulate* del grupo PCF *TraceRoute* se establece como MQROUTE_ACCUMULATE_AND_REPLY, y la siguiente actividad que debe realizarse en el mensaje de ruta de rastreo:

- es una operación de descartar
- es una operación de transferencia a una cola local (cola de destino o cola de mensajes no entregados)
- hará que el número total de actividades realizadas en el mensaje de ruta de rastreo supere el valor del parámetro *MaxActivities*, en el grupo PCF *TraceRoute*.

se genera un mensaje de respuesta de ruta de rastreo, y se entrega en la cola de respuesta especificada en el descriptor de mensajes del mensaje de ruta de rastreo.

COLA

El gestor de colas está habilitado para la mensajería de ruta de rastreo. Las aplicaciones dentro del ámbito del gestor de colas pueden grabar información de actividad en el mensaje de ruta de rastreo.

Si el parámetro *Accumulate* del grupo PCF *TraceRoute* se establece como MQROUTE_ACCUMULATE_AND_REPLY, y la siguiente actividad que debe realizarse en el mensaje de ruta de rastreo:

- es una operación de descartar
- es una operación de transferencia a una cola local (cola de destino o cola de mensajes no entregados)
- hará que el número total de actividades realizadas en el mensaje de ruta de rastreo supere el valor del parámetro *MaxActivities*, en el grupo PCF *TraceRoute*.

se genera un mensaje de respuesta de ruta de rastreo, y se entrega en la cola del sistema local SYSTEM.ADMIN.TRACE.ROUTE.QUEUE.

DISABLED

El gestor de colas está inhabilitado para la mensajería de ruta de rastreo. La información de actividad no se acumula en el mensaje de ruta de rastreo, sin embargo el grupo PCF *TraceRoute* se puede actualizar mientras se encuentre en el ámbito de este gestor de colas.

Por ejemplo, para inhabilitar la mensajería de ruta de rastreo en un gestor de colas, utilice el mandato MQSC siguiente:

```
ALTER QMGR ROUTEREC(DISABLED)
```

Recuerde: Cuando modifique el atributo de gestor de colas *ROUTEREC*, un MCA que esté en ejecución no detectará el cambio efectuado hasta que se reinicie el canal.

Habilitación de aplicaciones para la mensajería de ruta de rastreo

Para habilitar la mensajería de ruta de rastreo para una aplicación de usuario, base su algoritmo en el algoritmo que utilicen los agentes de canal de mensajes (MCA).

Antes de empezar

Si no está familiarizado con el formato de un mensaje de ruta de rastreo, consulte [“Referencia de mensaje de ruta de rastreo”](#) en la página 127.

Acerca de esta tarea

Los agentes de canal de mensajes (MCA) están habilitados para la mensajería de ruta de rastreo. Para habilitar una aplicación de usuario para la mensajería de ruta de rastreo, utilice los pasos siguientes del algoritmo que los MCA utilizan:

Procedimiento

1. Determine si el mensaje que se procesa es un mensaje de ruta de rastreo.
Si el mensaje no se ajusta al formato de un mensaje de ruta de rastreo, el mensaje no se procesa como si fuera uno de este tipo.
2. Determine si se debe registrar la información de actividad.
Si el nivel de detalle de la actividad realizada no es menor que el nivel de detalle especificado por el parámetro *Detail*, la información de actividad se registra bajo circunstancias específicas. Esta información sólo se registra si el mensaje de ruta de rastreo solicita la acumulación, y el gestor de colas está habilitado para la mensajería de ruta de rastreo, o si mensaje de ruta de rastreo solicita un informe de actividad y el gestor de colas se ha habilitado para el registro de actividad.
 - Si se debe registrar la información de actividad, incremente el parámetro *RecordedActivities*.
 - Si no se debe registrar la información de actividad, incremente el parámetro *UnrecordedActivities*.
3. Determine si el número total de actividades realizadas en el mensaje de ruta de rastreo excede el valor del parámetro *MaxActivities*.
El número total de actividades es la suma de *RecordedActivities*, *UnrecordedActivities* y *DiscontinuityCount*.
Si el número total de actividades excede el valor de *MaxActivities*, el mensaje se rechaza con el retorno MQFB_MAX_ACTIVITIES.
4. Si el valor de *Accumulate* se establece como MQROUTE_ACCUMULATE_IN_MSG o MQROUTE_ACCUMULATE_AND_REPLY, y el gestor de colas está habilitado para la mensajería de ruta de rastreo, escriba un grupo PCF Activity al final del bloque PCF en los datos de mensaje de un mensaje de ruta de rastreo.
5. Entregue el mensaje de ruta de rastreo en una cola local.
 - Si se especifica el parámetro *Deliver* como MQROUTE_DELIVER_NO, se rechaza el mensaje de ruta de rastreo con el retorno MQFB_NOT_DELIVERED.
 - Si se especifica el parámetro *Deliver* como MQROUTE_DELIVER_YES, se entrega el mensaje de ruta de rastreo en la cola local.
6. Genere un mensaje de respuesta de ruta de rastreo si son verdaderas todas las condiciones siguientes:

- El mensaje de ruta de rastreo se ha entregado en una cola local o se ha rechazado.
- El valor del parámetro *Accumulate* es MQROUTE_ACCUMULATE_AND_REPLY.
- El gestor de colas está habilitado para la mensajería de ruta de rastreo.

El mensaje de respuesta de ruta de rastreo se ha colocado en la cola determinada mediante el atributo de gestor de colas ROUTEREC.

7. Si el mensaje de ruta de rastreo ha solicitado un informe de actividad y el gestor de colas se ha habilitado para el registro de actividad, genere un informe de actividad.

El informe de actividad se coloca en la cola que se determina mediante el atributo de gestor de colas ACTIVREC.

Generación y configuración de un mensaje de ruta de rastreo

Un mensaje de ruta de rastreo consta del descriptor de mensaje específico y de los componentes de datos de mensaje. Para generar un mensaje de ruta de rastreo, cree el mensaje manualmente o utilice la aplicación de visualización de ruta de IBM MQ.

Un mensaje de ruta de rastreo consta de los componentes siguientes:

Descriptor de mensaje

Una estructura MQMD, con el campo *Formato* establecido en MQFMT_ADMIN o en MQFMT_EMBEDDED_PCF.

Datos de mensaje

Una de las combinaciones siguientes:

- Un cabecera PCF (MQCFH) y datos del mensaje de ruta de rastreo, si se ha establecido *Formato* en MQFMT.
- Una cabecera PCF incrustada (MQEPH), los datos del mensaje de ruta de rastreo, y datos de mensaje adicionales especificados por el usuario, si se ha establecido *Formato* en MQFMT_EMBEDDED_PCF.

Los datos de mensaje de ruta de rastreo constan del grupo PCF *TraceRoute* y uno o más grupos PCF *Activity*.

Generación manual

Cuando se genera un mensaje de ruta de rastreo manualmente, no se necesita ningún grupo PCF *Activity*. Los grupos PCF *Activity* se graban en los datos de mensaje del mensaje de ruta de rastreo cuando una aplicación escrita por el usuario, o una aplicación MCA, realiza una actividad en su nombre.

La aplicación de visualización de ruta de IBM MQ

La aplicación de visualización de ruta de IBM MQ, **dspmqrte**, se usa para configurar, generar y poner un mensaje de ruta de rastreo en una red de gestores de colas. Establezca el parámetro *Format* en el descriptor de mensaje en MQFMT_ADMIN. No puede añadir datos de usuario al mensaje de ruta de rastreo generado por la aplicación de visualización de ruta de IBM MQ.

Restricción: dspmqrte no se puede emitir en gestores de colas IBM MQ for z/OS . Si desea que el primer gestor de colas a través del que se direcciona el mensaje de ruta de rastreo sea un gestor de colas de este tipo, conéctese al gestor de colas como un cliente utilizando el parámetro opcional -c.

Imitación del mensaje original

Cuando utilice un mensaje de ruta de rastreo para determinar la ruta que haya seguido otro mensaje a través de una red de gestores de colas, cuanto mejor imite un mensaje de ruta de rastreo al mensaje original, más posibilidades tendrá de que el mensaje de ruta de rastreo siga la misma ruta que el mensaje original.

Las características de mensaje siguientes pueden afectar al lugar donde se reenvía un mensaje, dentro en una red de gestores de colas:

Prioridad

La prioridad puede especificarse en el descriptor de mensaje del mensaje.

Persistence

La persistencia puede especificarse en el descriptor de mensaje del mensaje.

Caducidad

La caducidad puede especificarse en el descriptor de mensaje del mensaje.

Opciones de informe

Las opciones de informe pueden especificarse en el descriptor de mensaje del mensaje.

Tamaño del mensaje

Para imitar el tamaño de un mensaje, se pueden grabar datos adicionales en los datos de mensaje del mensaje. Con este fin, los datos de mensaje adicionales pueden carecer de sentido.

Consejo: La aplicación de visualización de ruta de IBM MQ no puede especificar el tamaño de mensaje.

Datos de mensaje

Algunas redes de gestores de colas utilizan el direccionamiento basado en contenido para determinar a dónde se reenvían mensajes. En estos casos, los datos de mensaje del mensaje de ruta de rastreo se deben grabar para poder imitar los datos de mensaje del mensaje original.

Consejo: La aplicación de visualización de ruta de IBM MQ no puede especificar datos de mensaje.

Grupo PCF TraceRoute

Los atributos del grupo *PCF TraceRoute* controlan el comportamiento de un mensaje de ruta de rastreo. El grupo *TraceRoute* está en los datos de mensaje de cada mensaje de ruta de rastreo.

En la tabla siguiente se listan los parámetros del grupo *TraceRoute* que un MCA reconoce. Se pueden añadir más parámetros si se escriben aplicaciones escritas por el usuario para reconocerlas, tal como se describe en [“Información de actividad adicional”](#) en la página 80.

Parámetro	Tipo
TraceRoute	MQCFGR
Detail	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
Accumulate	MQCFIN
Forward	MQCFIN
Deliver	MQCFIN

A continuación figuran las descripciones de cada parámetro del grupo *PCF TraceRoute*:

Detail

Especifica el nivel de detalle de la información de actividad que se va a registrar. El valor puede ser cualquiera de los valores siguientes:

MQRROUTE_DETAIL_LOW

Sólo se registran las actividades realizadas por la aplicación de usuario.

MQRROUTE_DETAIL_MEDIUM

Se deberían registrar las actividades especificadas en **MQRROUTE_DETAIL_LOW**. Además, se registran las actividades realizadas por los MCA.

MQRROUTE_DETAIL_HIGH

Se deberían registrar las actividades especificadas en **MQRROUTE_DETAIL_LOW** y **MQRROUTE_DETAIL_MEDIUM**. Los MCA no registran ninguna otra información de actividad en este nivel de detalle. Esta opción sólo está disponible para las aplicaciones de usuario que vayan a registrar información de actividad adicional. Por ejemplo, si una aplicación de usuario

determina la ruta que toma un mensaje considerando ciertas características del mensaje, la lógica de direccionamiento podría incluirse en este nivel de detalle.

RecordedActivities

Especifica el número de actividades registradas que se pueden realizar en nombre del mensaje de ruta de rastreo. Se considera que se ha registrado una actividad si se ha escrito información acerca de la misma en el mensaje de ruta de rastreo, o se ha generado si un informe de actividad. Para cada actividad registrada, *RecordedActivities* se incrementa en uno.

UnrecordedActivities

Especifica el número de actividades no registradas que se pueden realizar en nombre del mensaje de ruta de rastreo. Se considera que una actividad se ha eliminado del registro si una aplicación que se haya habilitado para la mensajería de ruta de rastreo no acumula, ni graba, la información de actividad relacionada con un informe de actividad.

Una actividad realizada en nombre de un mensaje de ruta de rastreo se registrará en los casos siguientes:

- El nivel de detalle de la actividad realizada es menor que el nivel de detalle que especifica el parámetro *Detail*.
- El mensaje de ruta de rastreo solicita un informe de actividad, pero no la acumulación, y el gestor de colas no está habilitado para el registro de actividad.
- El mensaje de ruta de rastreo solicita la acumulación pero ningún informe de actividad, y el gestor de colas no está habilitado para la mensajería de ruta de rastreo.
- El mensaje de ruta de rastreo solicita tanto la acumulación como el informe de actividad, y el gestor de colas no está habilitado para el registro de actividad y la mensajería de ruta de rastreo.
- El mensaje de ruta de rastreo no solicita ni la acumulación ni ningún informe de actividad.

El parámetro *UnrecordedActivities* se incrementa en uno para cada actividad que no se haya registrado.

DiscontinuityCount

Especifica el número de veces que se ha direccionado el mensaje de ruta de rastreo a través de un gestor de colas con aplicaciones en las que no se había habilitado la mensajería de ruta de rastreo. El gestor de colas incrementa este valor. Si este valor es mayor que 0, sólo se puede determinar una ruta de mensaje parcial.

MaxActivities

Especifica el número máximo de actividades que se pueden realizar en nombre del mensaje de ruta de rastreo.

El número total de actividades es la suma de *RecordedActivities*, *UnrecordedActivities* y *DiscontinuityCount*. El número total de actividades no debe sobrepasar el valor de *MaxActivities*.

El valor de *MaxActivities* puede ser:

Un entero positivo

El número máximo de actividades.

Si se supera el número máximo de actividades, se rechaza el mensaje de ruta de rastreo con el retorno MQFB_MAX_ACTIVITIES. Esto puede impedir que el mensaje de ruta de rastreo se reenvíe indefinidamente si cae en un bucle infinito.

MQROUTE_UNLIMITED_ACTIVITIES

Se puede realizar un número ilimitado de actividades en nombre del mensaje de ruta de rastreo.

Accumulate

Especifica el método utilizado para acumular información de actividad. El valor puede ser cualquiera de los valores siguientes:

MQROUTE_ACCUMULATE_IN_MSG

Si el gestor de colas se ha habilitado para la mensajería de ruta de rastreo, la información de actividad se acumula en los datos de mensaje del mensaje de ruta de rastreo.

Si se especifica este valor, los datos del mensaje de ruta de rastreo constan de lo siguiente:

- El grupo PCF *TraceRoute*.
- Cero o más grupos PCF *Activity*.

MQRROUTE_ACCUMULATE_AND_REPLY

Si el gestor de colas se ha habilitado para la mensajería de ruta de rastreo, la información de actividad se acumula en los datos de mensaje del mensaje de ruta de rastreo, y se genera un mensaje de respuesta de ruta de rastreo si surge alguna de las situaciones siguientes:

- Un gestor de colas de IBM MQ descarta el mensaje de ruta de rastreo.
- Un gestor de colas de IBM MQ pone al mensaje de ruta de rastreo en una cola local (cola de destino o cola de mensajes no entregados).
- El número de actividades realizadas en el mensaje de ruta de rastreo es superior al valor de *MaxActivities*.

Si se especifica este valor, los datos del mensaje de ruta de rastreo constan de lo siguiente:

- El grupo PCF *TraceRoute*.
- Cero o más grupos PCF *Activity*.

MQRROUTE_ACCUMULATE_NONE

La información de actividad no se acumula en los datos de mensaje del mensaje de ruta de rastreo.

Si se especifica este valor, los datos del mensaje de ruta de rastreo constan de lo siguiente:

- El grupo PCF *TraceRoute*.

Forward

Especifica a dónde se puede reenviar un mensaje de ruta de rastreo. El valor puede ser:

MQRROUTE_FORWARD_IF_SUPPORTED

El mensaje de ruta de rastreo sólo se reenvía a los gestores de colas que acepten el valor del parámetro *Deliver* del grupo *TraceRoute*.

MQRROUTE_FORWARD_ALL

El mensaje de ruta de rastreo se reenvía a cualquier gestor de colas, independientemente de si se acepta el valor del parámetro *Deliver*.

Los gestores de colas utilizan el algoritmo siguiente al determinar si se debe reenviar un mensaje de ruta de rastreo un gestor de colas remoto:

1. Determine si el gestor de colas remoto es capaz de dar soporte a la mensajería de ruta de rastreo.
 - Si el gestor de colas remoto es capaz de dar soporte a la mensajería de ruta de rastreo, el algoritmo prosigue en el paso “4” en la [página 78](#).
 - Si el gestor de colas remoto no es capaz de dar soporte a la mensajería de ruta de rastreo, el algoritmo prosigue en el paso “2” en la [página 77](#).
2. Determine si el parámetro *Deliver* del grupo *TraceRoute* contiene alguna opción de entrega no reconocida en la máscara de bits MQRROUTE_DELIVER_REJ_UNSUP_MASK.
 - Si se encuentra alguna de las opciones de entrega no reconocidas, se rechaza el mensaje de ruta de rastreo con el retorno MQFB_UNSUPPORTED_DELIVERY.
 - Si no se encuentra ninguna opción de entrega no reconocida, el algoritmo prosigue en el paso “3” en la [página 77](#).
3. Determine el valor del parámetro *Deliver* del grupo PCF *TraceRoute* en el mensaje de ruta de rastreo.
 - Si se especifica *Deliver* como MQRROUTE_DELIVER_YES, el mensaje de ruta de rastreo se reenvía al gestor de colas remoto.
 - Si se especifica *Deliver* como MQRROUTE_DELIVER_NO, el algoritmo prosigue con el paso “4” en la [página 78](#).

4. Determine si el parámetro *Forward* del grupo *TraceRoute* contiene alguna opción de reenvío no reconocida en la máscara de bits `MQROUTE_FORWARDING_REJ_UNSUP_MASK`.
 - Si se encuentra alguna de las opciones de reenvío no reconocidas, se rechaza el mensaje de ruta de rastreo con el retorno `MQFB_UNSUPPORTED_FORWARDING`.
 - Si no se encuentra ninguna opción de reenvío no reconocida, el algoritmo prosigue en el paso [“5” en la página 78](#).
5. Determine el valor del parámetro *Forward* del grupo *PCF TraceRoute* en el mensaje de ruta de rastreo.
 - Si se especifica *Forward* como `MQROUTE_FORWARD_IF_SUPPORTED`, se rechaza el mensaje de ruta de rastreo con el retorno `MQFB_NOT_FORWARDED`.
 - Si se especifica *Forward* como `MQROUTE_FORWARD_ALL`, el mensaje de ruta de rastreo se reenvía al gestor de colas remoto.

Deliver

Especifica la acción que se debe efectuar si el mensaje de ruta de rastreo llega a su destino previsto. Las aplicaciones escritas por el usuario deben comprobar este atributo antes de colocar un mensaje de ruta de rastreo en su cola de destino. El valor puede ser cualquiera de los valores siguientes:

MQROUTE_DELIVER_YES

Al llegar, el mensaje de ruta de rastreo se coloca en la cola de destino. Cualquier aplicación que realice una operación de tipo `get` en la cola de destino puede recuperar el mensaje de ruta de rastreo.

MQROUTE_DELIVER_NO

Al llegar, el mensaje de ruta de rastreo no se entrega en la cola de destino. El mensaje se procesa según sus opciones de informe.

Configuración de una cola común para mensajes de respuesta de ruta de rastreo

Para determinar las ubicaciones de los mensajes de respuesta de ruta de rastreo relacionados con un mensaje específico cuando se entregan los informes a la cola del sistema local, resulta más eficaz utilizar una cola común en un único nodo.

Antes de empezar

Establezca el parámetro **ROUTEREC** para habilitar la mensajería de ruta de rastreo en el gestor de colas, y especifique que los mensajes de respuesta de ruta de rastreo generados se entreguen en la cola del sistema local `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`.

Acerca de esta tarea

Si se establece un número de gestores de colas en una red de gestores de colas para entregar los mensajes de respuesta de ruta de rastreo en la cola del sistema local, se puede tardar mucho tiempo en determinar las ubicaciones de los mensajes de respuesta de ruta de rastreo relacionados con un mensaje específico. De forma alternativa, utilice un único nodo, que es un gestor de colas en el que se aloja una cola común. Todos los gestores de colas de una red de gestores de colas pueden entregar los mensajes de respuesta de ruta de rastreo en esta cola común. La ventaja de utilizar una cola común es que los gestores de colas no tienen que entregar los mensajes de respuesta de ruta de rastreo en la cola de respuesta especificada en un mensaje y, al determinar las ubicaciones de los mensajes de respuesta de ruta de rastreo relacionados con un mensaje, sólo se consulta una cola.

Para configurar una cola común, efectúe los pasos siguientes:

Procedimiento

1. Seleccione o defina un gestor de colas como el único nodo.
2. En el único nodo, seleccione o defina una cola para utilizarla como la cola común

3. En todos los gestores de colas que reenvían los mensajes de respuesta de ruta de rastreo a la cola común, vuelva a definir la cola del sistema local SYSTEM.ADMIN.TRACE.ROUTE.QUEUE como una definición de cola remota.
 - a) Especifique el nombre del nodo único como el nombre del gestor de colas remoto.
 - b) Especifique el nombre de la cola común como el nombre de la cola remota.

Obtención y utilización de la información registrada

Utilice cualquiera de las técnicas siguientes para obtener la información de actividad registrada para un mensaje de ruta de rastreo.

Tenga en cuenta que las circunstancias en las que no se obtiene la información de actividad, también resultan aplicables a los mensajes de respuesta de ruta de rastreo.

La información de actividad no se graba cuando un gestor de colas, que esté inhabilitado para el registro de actividad y la mensajería de rastreo de ruta, procesa un mensaje de ruta de rastreo.

Adquisición de información de mensajes de respuesta de ruta de rastreo

Para obtener información de actividad, debe localizar el mensaje de respuesta de ruta de rastreo. A continuación, ya puede recuperar el mensaje, y analizar la información de actividad.

Acerca de esta tarea

Puede obtener información de actividad de un mensaje de respuesta de ruta de rastreo sólo si conoce la ubicación del mismo. Localice el mensaje y procese la información de actividad, tal como se indica a continuación:

Procedimiento

1. Compruebe la cola de respuesta que se haya especificado en el descriptor de mensaje del mensaje de ruta de rastreo. Si el mensaje de respuesta de ruta de rastreo no está en la cola de respuesta, compruebe las ubicaciones siguientes:
 - La cola del sistema local, SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, en el gestor de colas de destino del mensaje de ruta de rastreo.
 - La cola común, si ha configurado una cola común para los mensajes de respuesta de ruta de rastreo.
 - La cola del sistema local, SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, en cualquier otro gestor de colas en la red de gestores de colas, que puede producirse si el mensaje de ruta de rastreo se ha transferido a una cola de mensajes no entregados, o se ha superado el número máximo de actividades.
2. Recupere el mensaje de respuesta de ruta de rastreo.
3. Utilice la aplicación de visualización de ruta de IBM MQ para visualizar la información de actividad registrada.
4. Estudie la información de actividad y obtenga la información que necesite.

Adquisición de información de mensajes de ruta de rastreo

Para obtener información de actividad localice el mensaje de ruta de rastreo, que debe tener los parámetros adecuados en el grupo *TraceRoute*. A continuación, ya puede recuperar el mensaje, y analizar la información de actividad.

Acerca de esta tarea

Puede obtener información de actividad procedente de un mensaje de ruta de rastreo sólo si conoce la ubicación del mismo, y en éste se ha especificado el parámetro *Accumulate* en el grupo PCF *TraceRoute* como MQRROUTE_ACCUMULATE_IN_MSG o MQRROUTE_ACCUMULATE_AND_REPLY.

Para que el mensaje de ruta de rastreo se entregue en la cola de destino, se debe especificar el parámetro *Deliver* en el grupo PCF *TraceRoute* como MQRROUTE_DELIVER_YES.

Procedimiento

1. Compruebe la cola de destino. Si el mensaje de ruta de rastreo no está en la cola de destino, puede intentar localizarlo mediante un mensaje de ruta de rastreo en el que se haya habilitado el registro de actividad. Utilice los informes de actividad generados para tratar de determinar la última ubicación conocida del mensaje de ruta de rastreo.
2. Recupere el mensaje de ruta de rastreo.
3. Utilice la aplicación de visualización de ruta de IBM MQ para visualizar la información de actividad registrada.
4. Estudie la información de actividad y obtenga la información que necesite.

Adquisición de información de informes de actividad

Para obtener información de actividad localice el informe de actividad, que debe tener la opción de informe especificada en el descriptor de mensaje. A continuación, ya puede recuperar el informe de actividad, y analizar la información de actividad.

Acerca de esta tarea

Puede obtener información de actividad de un informe de actividad sólo si conoce la ubicación del mismo, y se ha especificado la opción de informe MQRO_ACTIVITY en el descriptor de mensaje del mensaje de ruta de rastreo.

Procedimiento

1. Localice y ordene los informes de actividad que se generen para un mensaje de ruta de rastreo.
Cuando haya localizado los informes de actividad, puede ordenarlos manualmente o utilizar la aplicación de visualización de ruta de IBM MQ para ordenar y visualizar la información de actividad automáticamente.
2. Estudie la información de actividad y obtenga la información que necesite.

Información de actividad adicional

A medida que un mensaje de ruta de rastreo se direcciona a través de una red de gestores de colas, las aplicaciones de usuario pueden registrar información adicional incluyendo uno o más parámetros PCF adicionales cuando escriben el grupo *Activity* en los datos de mensaje del mensaje de ruta de rastreo, o de un informe de actividad.

La información de actividad adicional puede ayudar a los administradores del sistema a identificar la ruta que ha tomado un mensaje de ruta de rastreo, o por qué se ha tomado dicha ruta.

Si utiliza la aplicación de visualización de ruta de IBM MQ para visualizar la información registrada para un mensaje de ruta de rastreo, los parámetros PCF adicionales sólo se pueden visualizar con un identificador numérico, a menos que la aplicación de visualización de ruta de IBM MQ reconozca el identificador de cada parámetro. Para poder reconocer un identificador de parámetro, se debe registrar información adicional utilizando los parámetros PCF siguientes. Incluya estos parámetros PCF en un lugar apropiado del grupo PCF *Activity*.

GroupName

<i>Tabla 13. Nombre de grupo</i>	
Descripción	Parámetros agrupados que especifican la información adicional.
Identificador	MQGACF_VALUE_NAMING.
Tipo de datos	MQCFGR
Parámetros del grupo	<i>ParameterName</i> <i>ParameterValue</i>

ParameterName

Tabla 14. Nombre de parámetro	
Descripción	Contiene el nombre que la aplicación de visualización de ruta de IBM MQ va a mostrar, que coloca el valor de <i>ParameterValue</i> en contexto.
Identificador	MQCA_VALUE_NAME.
Tipo de datos	MQCFST
Incluido en el grupo PCF:	<i>GroupName</i> .
Valor:	El nombre que se debe visualizar.

ParameterValue

Tabla 15. Valor del parámetro	
Descripción	Contiene el valor que va a visualizar la aplicación de ruta de visualización de IBM MQ.
Identificador:	El identificador de la estructura PCF la información adicional.
Tipo de datos:	El tipo de datos de la estructura PCF la información adicional.
Incluido en el grupo PCF:	<i>GroupName</i> .
Valor:	El valor que se debe visualizar.

Ejemplos de registro de información de actividad adicional

En los ejemplos siguientes se ilustra cómo una aplicación de usuario puede registrar información adicional al realizar una actividad en nombre de un mensaje de ruta de rastreo. En ambos ejemplos se utiliza la aplicación de visualización de ruta de IBM MQ para generar un mensaje de ruta de rastreo, y visualizar la información de actividad que se le devuelve.

Registro de información de actividad adicional: Ejemplo 1

Una aplicación de usuario registra información de actividad adicional en un formato en el que la aplicación de visualización de ruta de IBM MQ *no* reconoce el identificador de parámetro.

1. La aplicación de visualización de ruta de IBM MQ se utiliza para generar y colocar un mensaje de ruta de rastreo en una red de gestores de colas. Las opciones necesarias se establecen para solicitar lo siguiente:
 - La información de actividad se acumula en los datos de mensaje del mensaje de ruta de rastreo.
 - Al llegar a la cola de destino, el mensaje de ruta de rastreo se descarta y se genera un mensaje de respuesta de ruta de rastreo y se entrega a una cola de respuesta especificada.
 - Tras recibir el mensaje de respuesta de ruta de rastreo, la aplicación de visualización de ruta de IBM MQ muestra la información de actividad acumulada.

El mensaje de ruta de rastreo se coloca en la red de gestores de colas.

2. A medida que se direcciona el mensaje de ruta de rastreo a través de la red de gestores de colas, una aplicación de usuario, que se ha habilitado para la mensajería de ruta de rastreo, lleva a cabo una actividad de detalle en nombre del mensaje. Además de grabar la información de actividad estándar en el mensaje de ruta de rastreo, la aplicación de usuario graba el siguiente parámetro PCF al final del grupo Activity:

ColorValue

Identificador
65536

Tipo de datos

MQCFST

Valor

'Red'

Este parámetro PCF adicional proporciona más información sobre la actividad que se ha realizado, sin embargo, se escribe en un formato en el que la aplicación de visualización de ruta de IBM MQ *no* reconoce el identificador de parámetro.

- Los mensajes de ruta de rastreo llegan a la cola de destino y se devuelve un mensaje de respuesta de ruta de rastreo a la aplicación de visualización de ruta de IBM MQ. La información de actividad adicional se visualiza de la siguiente manera:

```
65536: 'Red'
```

La aplicación de visualización de ruta de IBM MQ no reconoce el identificador de parámetro del parámetro PCF y lo muestra como un valor numérico. El contexto de la información adicional no está claro.

Para ver un ejemplo de cuándo la aplicación de visualización de ruta de IBM MQ reconoce el identificador de parámetro del parámetro PCF, consulte [“Registro de información de actividad adicional: Ejemplo 2”](#) en la página 82.

Registro de información de actividad adicional: Ejemplo 2

Una aplicación de usuario registra información de actividad adicional en un formato donde el identificador de parámetro se reconoce mediante la aplicación de visualización de ruta IBM MQ.

- La aplicación de visualización de ruta de IBM MQ se utiliza para generar y colocar un mensaje de ruta de rastreo en una red de gestores de colas de la misma manera que en el [“Registro de información de actividad adicional: Ejemplo 1”](#) en la página 81.
- A medida que se direcciona el mensaje de ruta de rastreo a través de la red de gestores de colas, una aplicación de usuario, que se ha habilitado para la mensajería de ruta de rastreo, lleva a cabo una actividad de detalle en nombre del mensaje. Además de grabar la información de actividad estándar en el mensaje de ruta de rastreo, la aplicación de usuario graba los siguientes parámetros PCF al final del grupo Activity:

ColorInfo

<i>Tabla 16. Información de color</i>	
Descripción	Parámetros agrupados que especifican información sobre un color.
Identificador:	MQGACF_VALUE_NAMING.
Tipo de datos:	MQCFGR.
Parámetros del grupo:	<i>ColorName</i> <i>ColorValue</i>

ColorName

<i>Tabla 17. Nombre del color</i>	
Descripción	Contiene el nombre que va a visualizar la aplicación de ruta de visualización de IBM MQ que coloca el valor de <i>ColorValue</i> en contexto.
Identificador:	MQCA_VALUE_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>ColorInfo</i> .

Tabla 17. Nombre del color (continuación)	
Descripción	Contiene el nombre que va a visualizar la aplicación de ruta de visualización de IBM MQ que coloca el valor de <i>ColorValue</i> en contexto.
Valor:	'Color'

ColorValue

Tabla 18. Valor de color	
Descripción	Contiene el valor que va a visualizar la aplicación de ruta de visualización de IBM MQ.
Identificador:	65536.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>ColorInfo</i> .
Valor:	'Red'


Estos parámetros PCF adicionales proporcionan más información sobre la actividad que se ha realizado. Estos parámetros PCF se escriben en un formato donde el identificador de parámetro se reconoce mediante la aplicación de visualización de ruta IBM MQ.

- Los mensajes de ruta de rastreo llegan a la cola de destino y se devuelve un mensaje de respuesta de ruta de rastreo a la aplicación de visualización de ruta de IBM MQ. La información de actividad adicional se visualiza de la siguiente manera:

```
Color: 'Red'
```

La aplicación de visualización de ruta de IBM MQ reconoce que el identificador de parámetro de la estructura PCF que contiene el valor de la información de actividad adicional tiene un nombre correspondiente. El nombre correspondiente se muestra en lugar del valor numérico.

Aplicación de visualización de ruta de IBM MQ

Utilice la aplicación de visualización de ruta de IBM MQ (**dspmqrte**) para trabajar con los mensajes de ruta de rastreo y la información de actividad relacionada con un mensaje de ruta de rastreo, mediante una interfaz de línea de mandatos.  La aplicación de ruta de visualización de IBM MQ no se envía en IBM MQ for z/OS, pero puede ejecutarla desde una instalación distribuida y conectarla como cliente a un gestor de colas de IBM MQ for z/OS especificando el parámetro **-c** al emitir el mandato **dspmqrte**.

Puede utilizar la aplicación de visualización de ruta de IBM MQ [dspmqrte](#) para los fines siguientes:

- Configurar, generar y colocar un mensaje de ruta de rastreo en una red de gestores de colas.

Al colocar un mensaje de ruta de rastreo en una red de gestores de colas, se puede recoger la información de actividad, y puede utilizarse para determinar la ruta que ha tomado el mensaje de ruta de rastreo. Puede especificar las características de los mensajes de ruta de rastreo tal como se indica a continuación:

- El destino del mensaje de ruta de rastreo.
 - Cómo el mensaje de ruta de rastreo imita a otro mensaje.
 - Cómo debe manejarse el mensaje de ruta de rastreo mientras se direcciona a través de una red de gestores de colas.
 - Si se utiliza el registro de actividad, o la mensajería de ruta de rastreo, para registrar la información de actividad.
- Para ordenar y visualización la información de actividad relacionada con un mensaje de ruta de rastreo.

Si la aplicación de visualización de ruta de IBM MQ ha transferido un mensaje de ruta de rastreo a una red de gestores de colas, después de que se ha devuelto la información de actividad relacionada, la información se puede ordenar y mostrar inmediatamente. De forma alternativa, se puede utilizar la aplicación de visualización de ruta de IBM MQ para ordenar y visualizar la información de actividad relacionada con un mensaje de ruta de rastreo que se haya generado anteriormente.

Referencia relacionada

[dspmqrte](#)

Parámetros de los mensajes de ruta de rastreo

Utilice esta página para tener una visión general de los parámetros proporcionados por la aplicación de visualización de ruta de IBM MQ, **dspmqrte**, para determinar las características de un mensaje de ruta de rastreo, incluida la forma en que se maneja a medida que se direcciona a través de una red de gestores de colas.

Referencia relacionada

[dspmqrte](#)

Conexión del gestor de colas

Utilice esta página para especificar el gestor de colas al que se conecta la aplicación de visualización de ruta de IBM MQ.

-c

Especifica que la aplicación de visualización de ruta de IBM MQ se conecta como una aplicación cliente.

Si no especifica este parámetro, la aplicación de visualización de ruta de IBM MQ no se conecta como aplicación cliente.

-m NombreGestColas

El nombre del gestor de colas al que se conecta la aplicación de visualización de ruta de IBM MQ. El nombre puede tener hasta 48 caracteres.

Si no especifica este parámetro, se utiliza el gestor de colas predeterminado.

El destino señalado

Utilice esta página para especificar el destino objetivo de un mensaje de ruta de rastreo.

-q TargetQName

Si la aplicación de visualización de ruta de IBM MQ se está utilizando para enviar un mensaje de ruta de rastreo a una red de gestores de colas, *NombreColaDestino* especifica el nombre de la cola de destino.

-ts Serie_Tema_Destino

Especifica la serie del tema.

-qm GestorColasDestino

Califica el destino objetivo; entonces se aplicará la resolución de nombres de gestor de colas normal. El destino se especifica con *-q TargetQName* o *-ts TargetTopicString* .

Si no especifica este parámetro, el gestor de colas al que está conectada la aplicación de visualización de ruta de IBM MQ se utiliza como el gestor de colas de destino.

-o

Indica que el destino objetivo no está enlazado a un destino específico. Por lo general, este parámetro se utiliza cuando el mensaje de ruta de rastreo debe transferirse a través de un clúster. El destino objetivo se abre con la opción MQOO_BIND_NOT_FIXED.

Si no especifica este parámetro, el destino objetivo se enlaza a un destino específico.

Tema de publicación

Para las aplicaciones de publicación/suscripción, utilice esta página para especificar la serie de tema de un mensaje de ruta de rastreo para que la aplicación de visualización de ruta de IBM MQ lo publique.

-ts TopicName

Especifica una serie de tema para la que la aplicación de visualización de ruta de IBM MQ va a publicar un mensaje de ruta de rastreo, y pone esta aplicación en modalidad de tema. En esta modalidad, la aplicación realiza un rastreo de todos los mensajes resultantes de la solicitud de publicación.

También puede utilizar la aplicación de visualización de ruta de IBM MQ para visualizar los resultados de un informe de actividad generado para mensajes de publicación.

Imitación de mensajes

Utilice esta página para configurar un mensaje de ruta de rastreo que imite un mensaje, por ejemplo, cuando el mensaje original no ha llegado a su destino previsto.

Un uso de la mensajería de ruta de rastreo es ayudar a determinar la última ubicación conocida de un mensaje que no ha llegado a su destino previsto. La aplicación de visualización de ruta de IBM MQ proporciona parámetros que pueden ayudarle a configurar un mensaje de ruta de rastreo para imitar el mensaje original. Cuando imite un mensaje, puede utilizar los parámetros siguientes:

-l Persistence

Especifica la persistencia del mensaje de ruta de rastreo generado. Los posibles valores para *Persistence* son:

sí

El mensaje de ruta de rastreo generado es persistente. (MQPER_PERSISTENT).

No

El mensaje de ruta de rastreo generado **no** es persistente. (MQPER_NOT_PERSISTENT).

q

El mensaje de ruta de rastreo generado hereda su valor de persistencia del destino especificado en *-q NombreColaDestino* o *-ts SerieTemaDestino*. (MQPER_PERSISTENCE_AS_Q_DEF).

Un mensaje de respuesta de ruta de rastreo, o cualquier mensaje de informe, devuelto compartirá el mismo valor de persistencia que el mensaje de ruta de rastreo original.

Si *Persistencia* se especifica como **yes**, debe especificar el parámetro *-rq ColaRespuestas*. La cola de respuesta no debe resolverse en una cola dinámica temporal.

Si no especifica este parámetro, el mensaje de ruta de rastreo generado **no** es persistente.

-p Priority

Especifica la prioridad del mensaje de ruta de rastreo. El valor de *Prioridad* es mayor que o igual a 0, o MQPRI_PRIORITY_AS_Q_DEF. MQPRI_PRIORITY_AS_Q_DEF indica que el valor de prioridad se toma del destino especificado en *-q NombreColaDestino* o *-ts SerieTemaDestino*.

Si no especifica este parámetro, el valor de prioridad se toma del destino especificado por *-q NombreColaDestino* o *-ts SerieTemaDestino*.

-xs Expiry

Especifica el tiempo de caducidad, en segundos, del mensaje de ruta de rastreo.

Si no especifica este parámetro, el tiempo de caducidad se establece en 60 segundos.

-ro none |OpciónInforme

none

Especifica que no se ha establecido ninguna opción de informe.

ReportOption

Especifica las opciones de informe para el mensaje de ruta de rastreo. Se pueden especificar varias opciones de informe separándolas con una coma. Los valores posibles para *OpciónInforme* son:

activity

Se establece la opción de informe MQRO_ACTIVITY.

coa

Se establece la opción de informe MQRO_COA_WITH_FULL_DATA.

cod

Se establece la opción de informe MQRO_COD_WITH_FULL_DATA.

exception

Se establece la opción de informe MQRO_EXCEPTION_WITH_FULL_DATA.

expiration

Se establece la opción de informe MQRO_EXPIRATION_WITH_FULL_DATA.

discard

Se establece la opción de informe MQRO_DISCARD_MSG.

Si no se especifica *-ro ReportOption* ni *-ro none*, se especifican las opciones de informe MQRO_ACTIVITY y MQRO_DISCARD_MSG.

La aplicación de visualización de ruta de IBM MQ no permite añadir datos de usuario al mensaje de ruta de rastreo. Si necesita añadir datos de usuario al mensaje de ruta de rastreo, debe generar el mensaje de ruta de rastreo manualmente.

Información de actividad registrada

Utilice esta página para especificar el método utilizado para devolver la información de actividad registrada, que después puede utilizar para determinar la ruta que ha seguido un mensaje de ruta de rastreo.

La información de actividad registrada se puede devolver tal como se indica a continuación:

- En informes de actividad
- En un mensaje de respuesta de ruta de rastreo
- En el mismo mensaje de ruta de rastreo (que se haya colocado en la cola de destino)

Cuando utilice **dspmqrte**, el método utilizado para devolver la información de actividad registrada se determina mediante los parámetros siguientes:

-ro actividad

Especifica que la información de actividad se devuelve mediante los informes de actividad. De forma predeterminada, el registro de actividad está habilitado.

-ac -ar

Especifica que la información de actividad se acumula en el mensaje de ruta de rastreo, y que se generará un mensaje de respuesta de ruta de rastreo.

-ac

Especifica que la información de actividad se debe acumular en el mensaje de ruta de rastreo.

Si no especifica este parámetro, la información de actividad no se acumula en el mensaje de rastreo de ruta.

-ar

Solicita que se genere un mensaje de respuesta de ruta de rastreo que contenga toda la información de actividad acumulada, en las siguientes situaciones:

- Un gestor de colas de IBM MQ descarta el mensaje de ruta de rastreo.
- Un gestor de colas de IBM MQ pone al mensaje de ruta de rastreo en una cola local (cola de destino o cola de mensajes no entregados).
- El número de actividades realizadas en el mensaje de ruta de rastreo es superior al valor especificado en *-s Actividades*.

-ac -d yes

Especifica que la información de actividad se acumula en el mensaje de ruta de rastreo, y que a la llegada, el mensaje de ruta de rastreo se colocará en la cola de destino.

-ac

Especifica que la información de actividad se debe acumular en el mensaje de ruta de rastreo.

Si no especifica este parámetro, la información de actividad no se acumula en el mensaje de rastreo de ruta.

-d yes

Al llegar, el mensaje de ruta de rastreo se coloca en la cola de destino, aunque el gestor de colas no admita la mensajería de ruta de rastreo.

Si no especifica este parámetro, el mensaje de ruta de rastreo no se coloca en la cola de destino.

El mensaje de ruta de rastreo se puede recuperar de la cola de destino, y se puede adquirir la información de actividad registrada.

Puede combinar estos métodos según convenga.

Además, el nivel de detalle de la información de actividad registrada se puede especificar mediante el parámetro siguiente:

-t Detalle

Especifica las actividades que se registran. Los valores posibles para *Detalle* son:

low

Sólo se registran las actividades realizadas por una aplicación definida por el usuario.

medium

Se registran las actividades especificadas en low. Además, se registran las actividades de publicación y las actividades realizadas por los MCA.

alto

Se registran las actividades especificadas en low y medium. Los MCA no exponen ninguna otra información de actividad en este nivel de detalle. Esta opción está disponible para aplicaciones definidas por el usuario que sólo van a exponer información de actividad adicional. Por ejemplo, si una aplicación definida por el usuario determina la ruta que toma un mensaje considerando ciertas características del mensaje, la lógica de direccionamiento podría incluirse en este nivel de detalle.

Si no especifica este parámetro, se registran las actividades de nivel medio.

De forma predeterminada la aplicación de visualización de ruta de IBM MQ utiliza una cola dinámica temporal para almacenar los mensajes devueltos. Cuando la aplicación de visualización de ruta de IBM MQ finaliza, la cola dinámica temporal se cierra y los mensajes se purgan. Si los mensajes devueltos son necesarios más allá del final de la ejecución actual de la aplicación de visualización de ruta de IBM MQ, deberá especificarse una cola permanente mediante los parámetros siguientes:

-rq ColaRespuestas

Especifica el nombre de cola de respuesta a la que se envían todas las respuestas al mensaje de ruta de rastreo. Si el mensaje de ruta de rastreo es persistente, o si se especifica el parámetro *-n*, debe especificarse una cola de respuesta que no sea una cola dinámica temporal.

Si no especifica este parámetro, se crea una cola de respuesta dinámica utilizando la cola modelo predeterminada del sistema, SYSTEM.DEFAULT.MODEL.QUEUE.

-rqm GestorColasResp

Especifica el nombre del gestor de colas donde reside la cola de respuesta. El nombre puede tener hasta 48 caracteres.

Si no especifica este parámetro, el gestor de colas al que está conectada la aplicación de visualización de ruta de IBM MQ se utiliza como el gestor de colas de respuesta.

Cómo se maneja el mensaje de ruta de rastreo

Utilice esta página para controlar cómo se maneja un mensaje de ruta de rastreo a medida que se direcciona a través de una red de gestores de colas.

Los parámetros siguientes pueden restringir a dónde se puede direccionar el mensaje de ruta de rastreo en la red de gestores de colas:

-d Deliver

Especifica si el mensaje de ruta de rastreo se ha de entregar en la cola de destino al llegar. Los posibles valores para *Entregar* son:

sí

Al llegar, el mensaje de ruta de rastreo se coloca en la cola de destino, aunque el gestor de colas no admita la mensajería de ruta de rastreo.

No

Al llegar, el mensaje de ruta de rastreo no se coloca en la cola de destino.

Si no especifica este parámetro, el mensaje de ruta de rastreo no se coloca en la cola de destino.

-f Reenviar

Especifica el tipo de gestor de colas al que se puede reenviar el mensaje de ruta de rastreo. Para obtener detalles del algoritmo que los gestores de colas utilizan para determinar si reenviar un mensaje a un gestor de colas remoto, consulte [“Grupo PCF TraceRoute”](#) en la [página 75](#). Los valores posibles para *Reenviar* son:

Todos

El mensaje de ruta de rastreo se reenvía a cualquier gestor de colas.

Aviso: Si se reenvía a un gestor de colas anterior a IBM WebSphere MQ 6.0, el mensaje de ruta de rastreo no se reconocerá y puede entregarse en una cola local, a pesar del valor del parámetro *-d Deliver*.

supported

El mensaje de ruta de rastreo solo se reenvía a un gestor de colas que acepte el parámetro *Entregar* del grupo PCF *RastreoRuta*.

Si no especifica este parámetro, el mensaje de ruta de rastreo sólo se reenviará a un gestor de colas que acepte el parámetro *Entregar*.

Los siguientes parámetros pueden impedir que un mensaje de ruta de rastreo permanezca en una red de gestores de colas de forma indefinida:

-s Actividades

Especifica el número máximo de actividades registradas que se pueden realizar en nombre del mensaje de ruta de rastreo antes de que sea descartado. Esto impide que el mensaje de ruta de rastreo se reenvíe indefinidamente si cae en un bucle infinito. El valor de *Actividades* es mayor que o igual a 1, o MQROUTE_UNLIMITED_ACTIVITIES. MQROUTE_UNLIMITED_ACTIVITIES especifica que se puede realizar un número ilimitado de actividades en nombre del mensaje de ruta de rastreo.

Si no especifica este parámetro, se puede realizar un número ilimitado de actividades en nombre del mensaje de ruta de rastreo.

-xs Expiry

Especifica el tiempo de caducidad, en segundos, del mensaje de ruta de rastreo.

Si no especifica este parámetro, el tiempo de caducidad se establece en 60 segundos.

-xp PasarCaducidad

Especifica si el tiempo de caducidad del mensaje de ruta de rastreo se pasa a un mensaje de respuesta de ruta de rastreo. Los posibles valores para *PasarCaducidad* son:

sí

La opción de informe MQRO_PASS_DISCARD_AND_EXPIRY se especifica en el descriptor de mensaje del mensaje de ruta de rastreo.

Si se genera un mensaje de respuesta de ruta de rastreo, o informes de actividad, para el mensaje de ruta de rastreo, la opción de informe MQRO_DISCARD (si se especifica) y el tiempo de caducidad restante se pasan en el mensaje.

Éste es el valor predeterminado.

No

La opción de informe MQRO_PASS_DISCARD_AND_EXPIRY no se especifica.

Si se genera un mensaje de respuesta de ruta de rastreo para el mensaje de ruta de rastreo, la opción discard y el tiempo de caducidad del mensaje de ruta de rastreo **no** se pasan en el mensaje.

Si no especifica este parámetro, no se especifica MQRO_PASS_DISCARD_AND_EXPIRY.

-ro discard

Especifica la opción de informe MQRO_DISCARD_MSG. Esto puede impedir que el mensaje de ruta de rastreo permanezca en la red de gestores de colas de forma indefinida.

Visualización de la información de actividad

La aplicación de visualización de ruta de IBM MQ puede mostrar información de actividad para un mensaje de ruta de rastreo que se acaba de colocar en una red de gestores de colas, o puede mostrar información de actividad para un mensaje de ruta de rastreo generado previamente. También puede mostrar información adicional que hayan registrado las aplicaciones escritas por el usuario.

Para especificar si se devuelve la información de actividad de un mensaje de ruta de rastreo, especifique el parámetro siguiente:

-n

Especifica que la información de actividad devuelta para el mensaje de ruta de rastreo no se debe mostrar.

Si este parámetro va acompañado de una solicitud para un mensaje de respuesta de ruta de rastreo, (*-ar*), o cualquiera de las opciones de generación de informes a partir de (*-ro ReportOption*), se debe especificar una cola de respuesta específica (no de modelo) utilizando *-rq ReplyToQ* . De forma predeterminada, sólo se solicitan mensajes de informe de actividad.

Cuando el mensaje de ruta de rastreo se ha colocado en la cola de destino especificada, se muestra una serie hexadecimal de 48 caracteres que contiene el identificador de mensaje del mensaje de ruta de rastreo. El identificador de mensaje lo puede utilizar la aplicación de visualización de ruta de IBM MQ para mostrar la información de actividad para el mensaje de ruta de rastreo en un momento posterior, mediante el parámetro *-i IDCORREL*.

Si no especifica este parámetro, la información de actividad devuelta para el mensaje de ruta de rastreo se visualiza en el formato especificado en el parámetro *-v*.

Cuando se visualiza la información de actividad de un mensaje de ruta de rastreo que se acaba de colocar en una red de gestores de colas, se puede especificar el parámetro siguiente:

-w TiempoEspera

Especifica el tiempo, en segundos, que la aplicación de visualización de ruta de IBM MQ esperará a que los informes de actividad, o un mensaje de ruta de rastreo, vuelvan a la cola de respuesta especificada.

Si no especifica este parámetro, el tiempo de espera se especifica como el tiempo de caducidad del mensaje de ruta de rastreo, más 60 segundos.

Cuando se visualiza la información de actividad acumulada previamente, se deben establecer los parámetros siguientes:

-q TargetQName

Si la aplicación de visualización de ruta de IBM MQ se está utilizando para ver información de actividad reunida previamente, *TargetQName* especifica el nombre de la cola donde se almacena la información de actividad.

-i IDCORREL

Este parámetro se utiliza cuando la aplicación de visualización de ruta de IBM MQ se utiliza para ver sólo información de actividad acumulada previamente. Puede haber muchos informes de actividad y mensajes de respuesta de rastreo de ruta en la cola especificada en *-q NombColaDestino*. *IDCORREL* se utiliza para identificar los informes de actividad, o un mensaje de respuesta de rastreo de ruta, relacionados con un mensaje de ruta de rastreo. Especifique el identificador de mensaje del mensaje de ruta de rastreo original en *IDCORREL*.

El formato de *IDCORREL* es una serie hexadecimal de 48 caracteres.

Los parámetros siguientes pueden utilizarse al visualizar la información de actividad acumulada previamente, o cuando se muestra la información de actividad actual de un mensaje de ruta de rastreo:

-b

Especifica que la aplicación de visualización de ruta de IBM MQ sólo examinará informes de actividad o un mensaje de respuesta de ruta de rastreo relacionado con un mensaje. Esto permite que la información de actividad se vuelva a visualizar más adelante.

Si no especifica este parámetro, la aplicación de visualización de ruta de IBM MQ obtendrá de forma destructiva informes de actividad o un mensaje de respuesta de ruta de rastreo relacionado con un mensaje.

-v summary | all | none | outline *OpciónVisualizar***resumen**

Se visualizan las colas a través de las cuales se ha direccionado el mensaje de ruta de rastreo.

Todos

Se visualiza toda la información disponible.

Ninguno

No se visualiza ninguna información.

outline *DisplayOption*

Especifica las opciones de visualización para el mensaje de ruta de rastreo. Se pueden especificar varias opciones de visualización utilizando la coma como separador.

Si no se proporciona ningún valor, se visualiza lo siguiente:

- El nombre de la aplicación
- El tipo de cada operación
- Todos los parámetros específicos de la operación

Los valores posibles para *OpciónVisualización* son:

activity

Se visualizan todos los parámetros de grupo no PCF en los grupos PCF *Actividad*.

identificadores

Se visualizan los valores que tienen los identificadores de parámetro MQBACF_MSG_ID o MQBACF_CORREL_ID. Este valor prevalece sobre *msgdelta*.

mensaje

Se visualizan todos los parámetros de grupo no PCF en los grupos PCF *Mensaje*. Cuando se especifica este valor, no se puede especificar *msgdelta*.

msgdelta

Se visualizan todos los parámetros de grupo no PCF en los grupos PCF *Mensaje* que han cambiado desde la última operación. Cuando se especifica este valor, no se puede especificar *message*.

operación

Se visualizan todos los parámetros de grupo no PCF en los grupos PCF de *Operation*.

traceroute

Se visualizan todos los parámetros de grupo no PCF en los grupos PCF *TraceRoute*.

Si no especifica este parámetro, se visualiza un resumen de la ruta de mensaje.

Visualización de la información adicional

A medida que un mensaje de ruta de rastreo se direcciona a través de una red de gestores de colas, las aplicaciones escritas por el usuario pueden registrar información adicional grabando uno o más parámetros PCF adicionales en los datos de mensaje del mensaje de ruta de rastreo, o en los datos de mensaje de un informe de actividad. Para que la aplicación de visualización de ruta de IBM MQ muestre información adicional en un formato legible, ésta debe registrarse en un formato específico, como se describe en [“Información de actividad adicional” en la página 80](#).

Ejemplos de la aplicación de visualización de ruta de IBM MQ

Los siguientes ejemplos muestran cómo puede utilizar la aplicación de visualización de ruta de IBM MQ. En cada ejemplo hay dos gestores de colas (QM1 y QM2) que están interconectados por dos canales (QM2.TO.QM1 y QM1.TO.QM2).

Ejemplo 1 - Solicitar informes de actividad

Visualizar la información de actividad de un mensaje de ruta de rastreo que se entrega en la cola de destino.

En este ejemplo, la aplicación de visualización de ruta de IBM MQ se conecta al gestor de colas QM1, y se utiliza para generar y entregar un mensaje de ruta de rastreo a la cola de destino TARGET.Q en el gestor de colas QM2. Se especifica la opción de informe necesaria, de modo que se soliciten los informes de actividad mientras se direcciona el mensaje de respuesta de rastreo de ruta. El mensaje de ruta de rastreo se descarta cuando llega a la cola de destino. La información de actividad que se devuelve a la aplicación de visualización de ruta de IBM MQ mediante los informes de actividad se ordena y se visualiza.

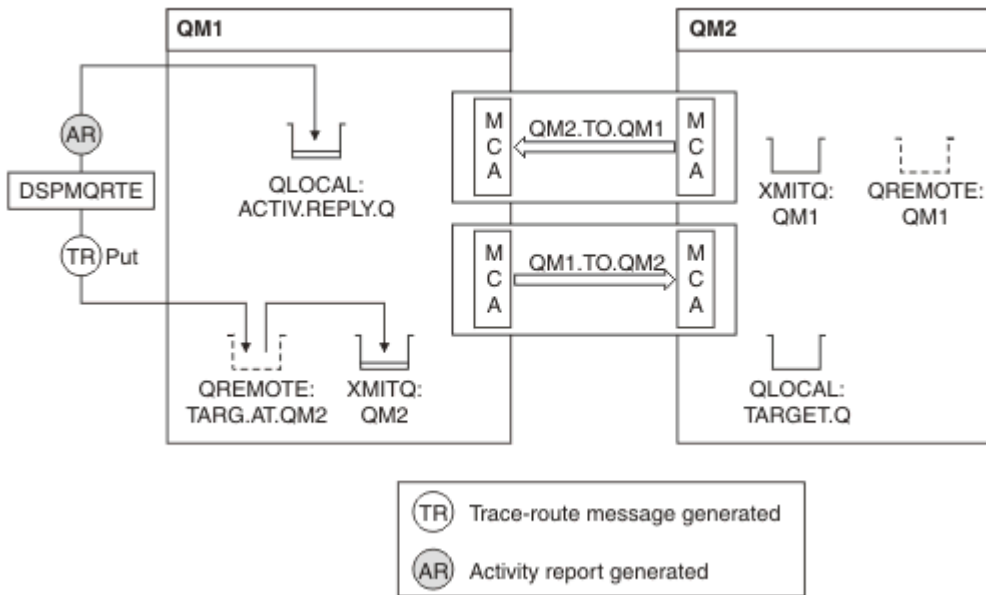


Figura 9. Solicitar informes de actividad, Diagrama 1

- El atributo ACTIVREC de cada gestor de colas (QM1 y QM2) se establece en MSG.
- El se emite mandato siguiente:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1 es el nombre del gestor de colas al que se conecta la aplicación de visualización de ruta de IBM MQ, TARG.AT.QM2 es el nombre de la cola de destino y ACTIV.REPLY.Q es el nombre de la cola a la que se solicita que se envíen todas las respuestas al mensaje de ruta de rastreo.

Se presuponen los valores predeterminados para todas las opciones que no se especifiquen, pero tenga en cuenta, en concreto, la opción -f (el mensaje de ruta de rastreo sólo se reenvía a un gestor de colas que acepte el parámetro Deliver del grupo PCF TraceRoute), la opción -d (al llegar, el mensaje de ruta de rastreo no se coloca en la cola de destino), la opción -ro (se especifican las opciones de informe MQRO_ACTIVITY y MQRO_DISCARD_MSG) y la opción -t (se graba la actividad de nivel de detalle medio).

- DSPMQRTE genera el mensaje de ruta de rastreo, y lo coloca en la cola TARG.AT.QM2.
- A continuación, DSPMQRTE, busca el valor del atributo ACTIVREC del gestor de colas QM1. El valor es MSG, por lo que DSPMQRTE genera un informe de actividad, y lo coloca en la cola de respuesta ACTIV.REPLY.Q.

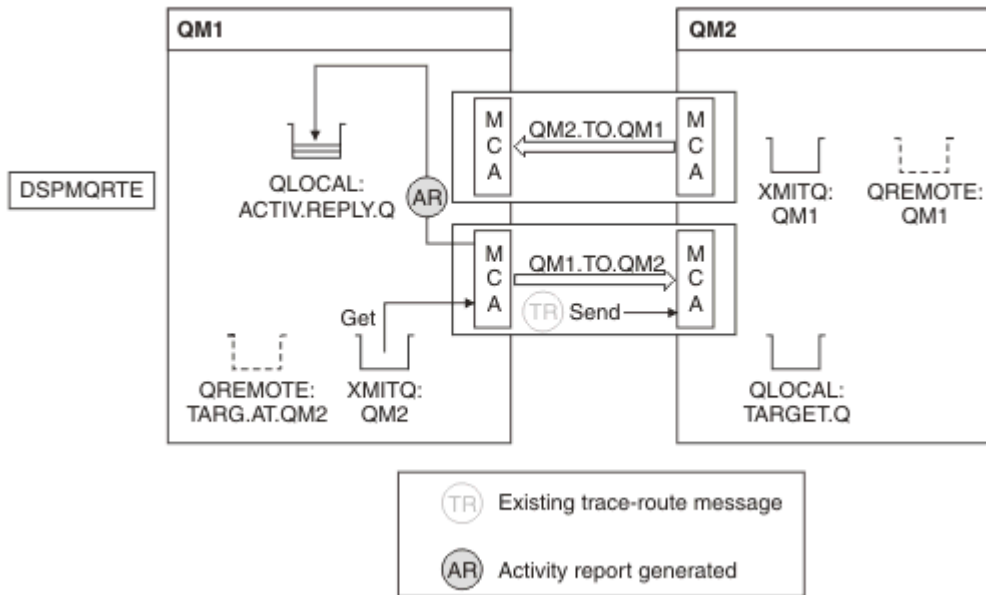


Figura 10. Solicitar informes de actividad, Diagrama 2

- El agente de canal de mensajes (MCA) de envío obtiene el mensaje de ruta de rastreo de la cola de transmisión. El mensaje es un mensaje de ruta de rastreo y, por tanto, el MCA empieza a registrar la información de actividad.
- El atributo ACTIVREC del gestor de colas (QM1) es MSG, y se especifica la opción MQRO_ACTIVITY en el campo Report del descriptor de mensaje, por lo que el MCA generará un informe de actividad más adelante. El valor del parámetro RecordedActivities del grupo PCF TraceRoute se incrementa en 1.
- El MCA comprueba que no se haya superado el valor MaxActivities del grupo PCF TraceRoute.
- Antes de que el mensaje se reenvíe a QM2, el MCA sigue el algoritmo que se describe en el parámetro Forward (los pasos “1” en la página 77, “4” en la página 78 y “5” en la página 78), y el MCA elige enviar el mensaje.
- A continuación, el MCA genera un informe de actividad y lo coloca en la cola de respuesta (ACTIV.REPLY.Q).

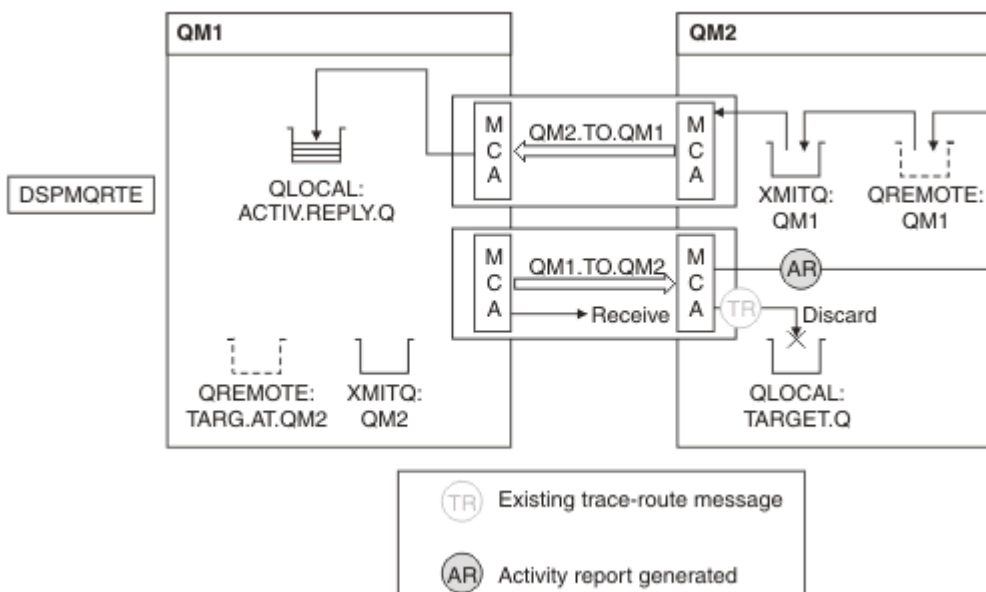


Figura 11. Solicitar informes de actividad, Diagrama 3

- El MCA receptor recibe el mensaje de ruta de rastreo del canal. El mensaje es un mensaje de ruta de rastreo y, por tanto, el MCA empieza a registrar la información sobre la actividad.
- Si el gestor de colas del cual procede el mensaje de ruta de rastreo es de IBM WebSphere MQ 5.3.1 o anterior, el MCA incrementa en 1 el parámetro DiscontinuityCount del grupo PCF TraceRoute. Este no es el caso que aquí se trata.
- El atributo ACTIVREC del gestor de colas (QM2) es MSG, y se especifica la opción MQRO_ACTIVITY, por lo que el MCA generará un informe de actividad. El valor del parámetro RecordedActivities se incrementa en 1.
- La cola de destino es una cola local y, por tanto, el mensaje se descarta con el retorno MQFB_NOT_DELIVERED, de acuerdo con el valor del parámetro Deliver del grupo PCF TraceRoute.
- A continuación, el MCA genera el informe de actividad final y lo coloca en la cola de respuesta. Esto se resuelve en la cola de transmisión asociada al gestor de colas QM1, y el informe de actividad se devuelve al gestor de colas QM1 (ACTIV.REPLY.Q).

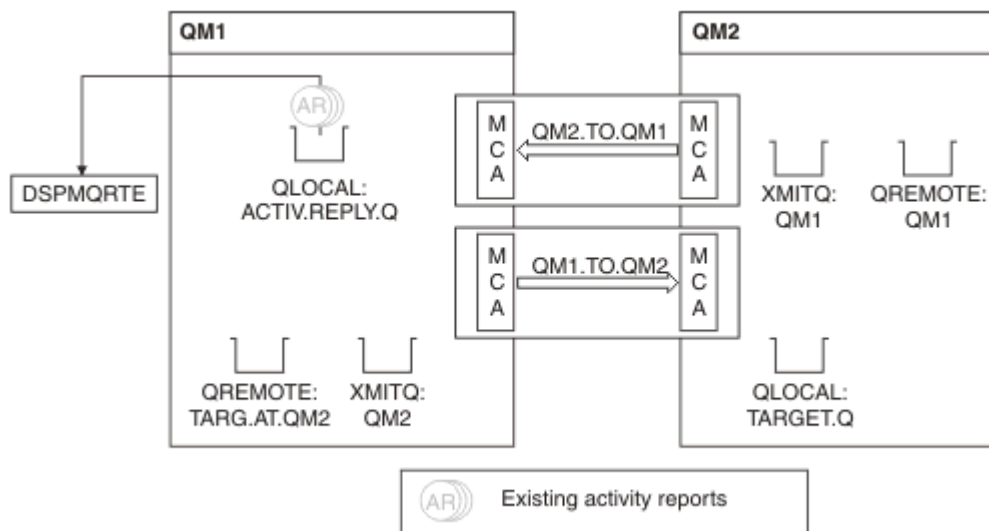


Figura 12. Solicitar informes de actividad, Diagrama 4

- Mientras, DSPMQRTE ha estado realizando llamadas MQGET continuamente en la cola de respuesta (ACTIV.REPLY.Q), a la espera de los informes de actividad. Esperará hasta 120 segundos (60 segundos más que el período de caducidad del mensaje de ruta de rastreo) ya que no se ha especificó -w al iniciar DSPMQRTE.
- DSPMQRTE obtiene los 3 informes de actividad desde fuera de la cola de respuesta.
- Los informes de actividad se ordenan mediante los parámetros RecordedActivities, UnrecordedActivities y DiscontinuityCount del grupo PCF TraceRoute, para cada una de las actividades. El único valor que no es cero en este ejemplo es RecordedActivities y, por tanto, es el único parámetro que se utiliza realmente.
- El programa finaliza en cuanto se muestra la operación Discard. Aunque la operación final fuera una operación Discard, se trata como si hubiera tenido lugar una transferencia, porque el retorno es MQFB_NOT_DELIVERED.

La salida que se muestra es la siguiente:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-iq ACTIV.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

Ejemplo 2 - Solicitar un mensaje de respuesta de ruta de rastreo
 Generar y entregar un mensaje de ruta de rastreo en la cola de destino.

En este ejemplo, la aplicación de visualización de ruta de IBM MQ se conecta al gestor de colas QM1, y se utiliza para generar y entregar un mensaje de ruta de rastreo a la cola de destino TARGET.Q en el gestor de colas QM2. Se especifica la opción necesaria, de modo que la información de actividad se acumule en el mensaje de ruta de rastreo. El mensaje de respuesta de ruta de rastreo se descarta cuando llega a la cola de destino, y se solicita un mensaje de respuesta de ruta de rastreo.

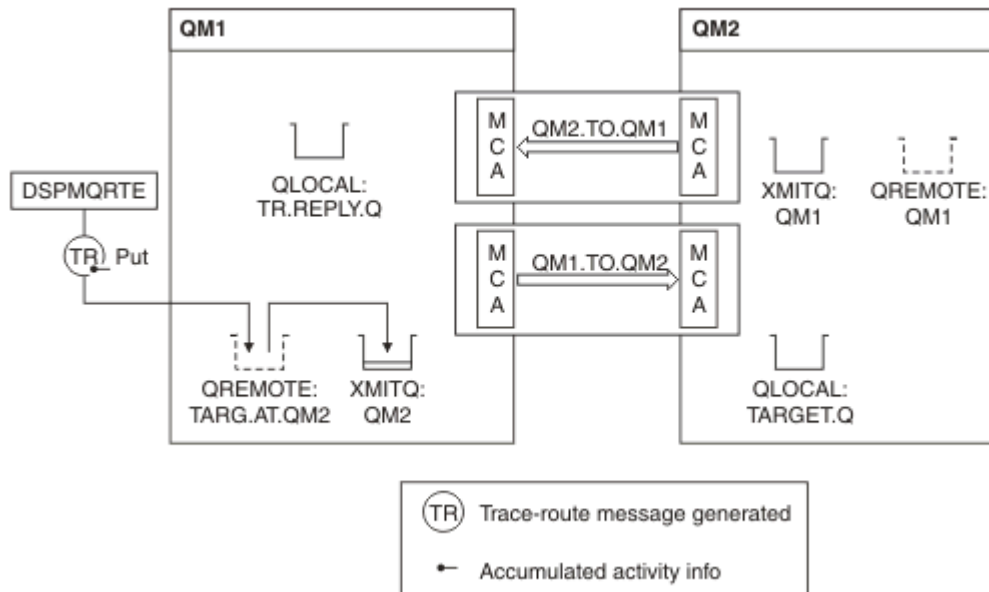


Figura 13. Solicitar un mensaje de respuesta de ruta de rastreo, Diagrama 1

- El atributo ROUTEREC de cada gestor de colas (QM1 y QM2) se establece en MSG.
- El se emite mandato siguiente:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq TR.REPLY.Q -ac -ar -ro discard
```

QM1 es el nombre del gestor de colas al que se conecta la aplicación de visualización de ruta de IBM MQ, TARG.AT.QM2 es el nombre de la cola de destino y ACTIV.REPLY.Q es el nombre de la cola a la que se solicita que se envíen todas las respuestas al mensaje de ruta de rastreo. La opción -ac especifica que la información de actividad se acumula en el mensaje de ruta de rastreo; la opción -ar especifica que toda la actividad acumulada se envíe a la cola de respuesta especificada con la opción -rq (es decir, TR.REPLY.Q). La opción -ro especifica que se establece la opción de informe MQRO_DISCARD_MSG, que significa que en este ejemplo no se generan informes de actividad.

- DSPMQRTE acumula la información de actividad en el mensaje de ruta de rastreo antes de que se transfiera el mensaje a la ruta de destino. El atributo de gestor de colas ROUTEREC no debe establecerse en DISABLED para que esto suceda.

- Ésta es la última actividad que se llevará a cabo en el mensaje y, debido a que el atributo de gestor de colas ROUTEREC en QM1 no se ha establecido en DISABLED, el MCA genera un mensaje de respuesta de ruta de rastreo de acuerdo con el valor de Accumulate. El valor de ROUTEREC es MSG y, por tanto, el mensaje de respuesta se transfiere a la cola de respuesta. El mensaje de respuesta contiene toda la información de actividad acumulada del mensaje de ruta de rastreo.

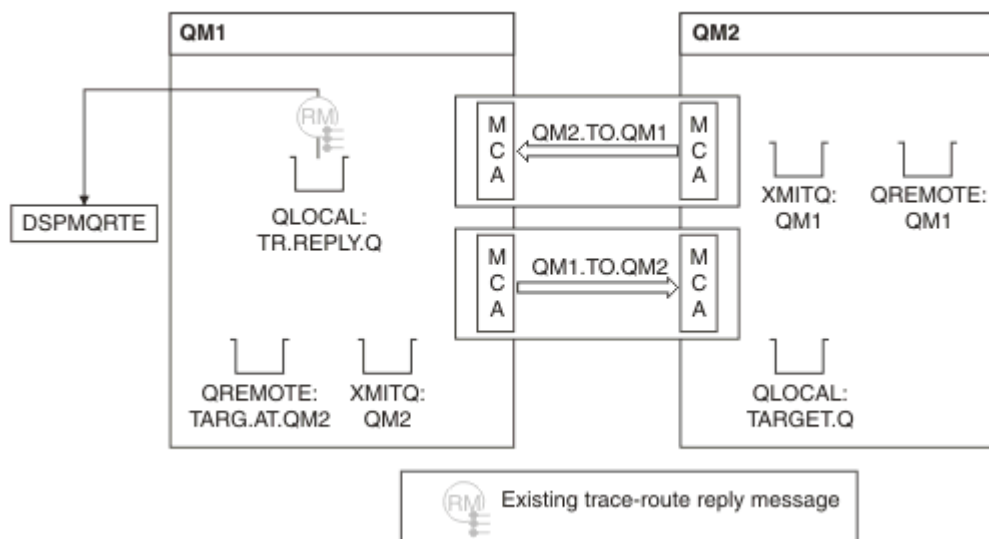


Figura 16. Solicitar un mensaje de respuesta de ruta de rastreo, Diagrama 4

- Mientras que DSPMQRTE espera a que el mensaje de respuesta de ruta de rastreo vuelva a la cola de respuesta. Cuando vuelve, DSPMQRTE analiza cada actividad que contiene y la imprime. La operación final es una operación de tipo discard. DSPMQRTE finaliza una vez se ha impreso.

La salida que se muestra es la siguiente:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
TR.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

Ejemplo 3 - Entrega de informes de actividad a la cola del sistema

Detectar cuándo se entregan informes de actividad a otras colas que no sean la cola de respuesta y utilizar la aplicación de visualización de ruta de IBM MQ para leer informes de actividad de la otra cola.

Este ejemplo es el mismo que el “Ejemplo 1 - Solicitar informes de actividad” en la página 91, excepto que ahora QM2 tiene el valor del atributo de gestor de colas ACTIVREC establecido en QUEUE. Debe haberse reiniciado el canal QM1.TO.QM2 para que esto surta efecto.

En este ejemplo se muestra cómo detectar cuándo se entregan informes de actividad a otras colas que no sea la cola de respuesta. Una vez se ha detectado, se utiliza la aplicación de visualización de ruta de IBM MQ para leer informes de actividad de otra cola.

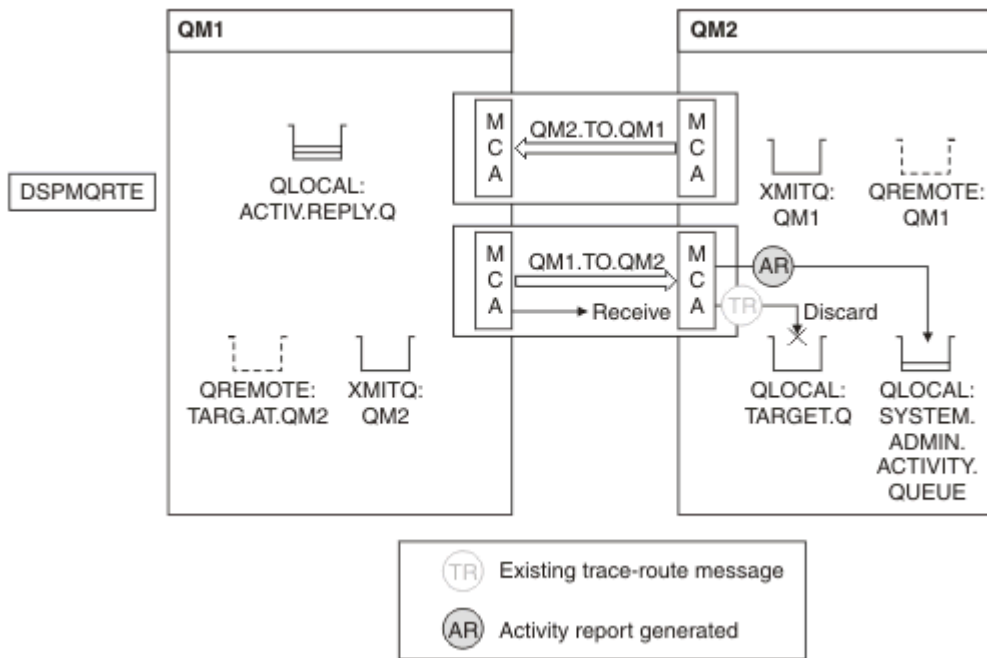


Figura 17. Entrega de informes de actividad a la cola del sistema, Diagrama 1

- El mensaje es un mensaje de ruta de rastreo y, por tanto, el MCA receptor empieza a registrar la información sobre la actividad.
- El valor del atributo de gestor de colas ACTIVREC en QM2 ahora es QUEUE y, por tanto, el MCA genera un informe de actividad, pero lo coloca en la cola del sistema (SYSTEM.ADMIN.ACTIVITY.QUEUE), y no en la cola de respuesta (ACTIV.REPLY.Q).

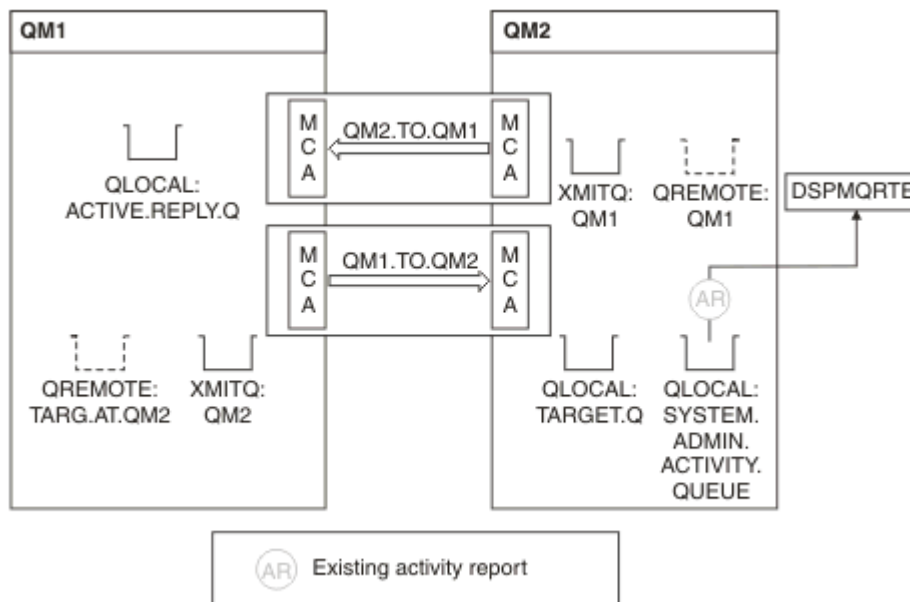


Figura 18. Entregar informes de actividad a la cola del sistema, Diagrama 2

- Mientras tanto, DSPMQRTE ha estado esperando que los informes de actividad lleguen a ACTIV.REPLY.Q. Solo llegan dos. DSPMQRTE sigue esperando 120 segundos porque parece que aún no se ha completado la ruta.

La salida que se muestra es la siguiente:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
ACTIV.REPLY.Q -v outline identifiers'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

```
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'
```

```
Operation:
  OperationType: Put
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
```

```
-----
Activity:
  ApplName: 'cann\output\bin\runmqchl.EXE'
```

```
Operation:
  OperationType: Get
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001505'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001502'
```

```
EmbeddedMQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'
```

```
Operation:
  OperationType: Send
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  RemoteQMgrName: 'QM2'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Sender
  XmitQName: 'QM2'
```

```
-----
AMQ8652: DSPMQRTE command has finished.
```

- La última operación que DSPMQRTE ha observado ha sido Send, por lo que el canal está en ejecución. Ahora tenemos que averiguar por qué no hemos recibido ningún informe de actividad más del gestor de colas QM2 (tal como se identifica en RemoteQMgrName).
- Para comprobar si existe alguna información de actividad en la cola del sistema, inicie DSPMQRTE en QM2 para intentarlo y recopilar más informes de actividad. Utilice el mandato siguiente para iniciar DSPMQRTE:

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

donde 414D51204C41524745512020202020A3C9154220001502 es el valor MsgId del mensaje de ruta de rastreo que se ha colocado.

- A continuación, DSPMQRTE vuelve a efectuar una secuencia de MQGET, esperando en la cola de la actividad del sistema las respuestas relacionadas con el mensaje de ruta de rastreo que tiene el identificador especificado.
- DSPMQRTE obtiene un informe de actividad más, el cual se visualiza. DSPMQRTE determina que faltan los informes de actividad anteriores, y hace que aparezca un mensaje en el que se explica esto. No obstante, ya conocemos esta parte de la ruta.

La salida que se muestra es la siguiente:

```

AMQ8653: DSPMQRTE command started with options '-m QM2
        -q SYSTEM.ADMIN.ACTIVITY.QUEUE
        -i 414D51204C41524745512020202020A3C915420001502 -v outline'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----

Activity:
  Activity information unavailable.
-----

Activity:
  ApplName: 'cann\output\bin\AMQRMPPA.EXE'

  Operation:
    OperationType: Receive
    QMgrName: 'QM2'
    RemoteQMGrName: 'QM1'
    ChannelName: 'QM1.TO.QM2'
    ChannelType: Receiver

  Operation:
    OperationType: Discard
    QMgrName: 'QM2'
    QName: 'TARGET.Q'
    Feedback: NotDelivered
-----

AMQ8652: DSPMQRTE command has finished.

```

- Este informe de actividad indica que la información de ruta está ahora completa. Se ha producido ningún problema.
- Sólo porque la información de ruta no esté disponible, o porque DSPMQRTE no pueda mostrar toda la ruta, no significa que el mensaje no se haya entregado. Por ejemplo, los atributos de gestor de colas de gestores de colas diferentes pueden ser diferentes, o es posible que no se haya definido una cola de respuesta para obtener la respuesta que se devuelve.

Ejemplo 4 - Diagnóstico de un problema de canal

Diagnosticar un problema en el que el mensaje de ruta de rastreo no llega a la cola de destino.

En este ejemplo, la aplicación de visualización de ruta de IBM MQ se conecta al gestor de colas QM1, genera un mensaje de ruta de rastreo y luego intenta entregarlo a la cola de destino TARGET.Q, en el gestor de colas remoto QM2. En este ejemplo, el mensaje de ruta de rastreo no llega a la cola de destino. Para diagnosticar el problema, se utiliza el informe de actividad disponible.

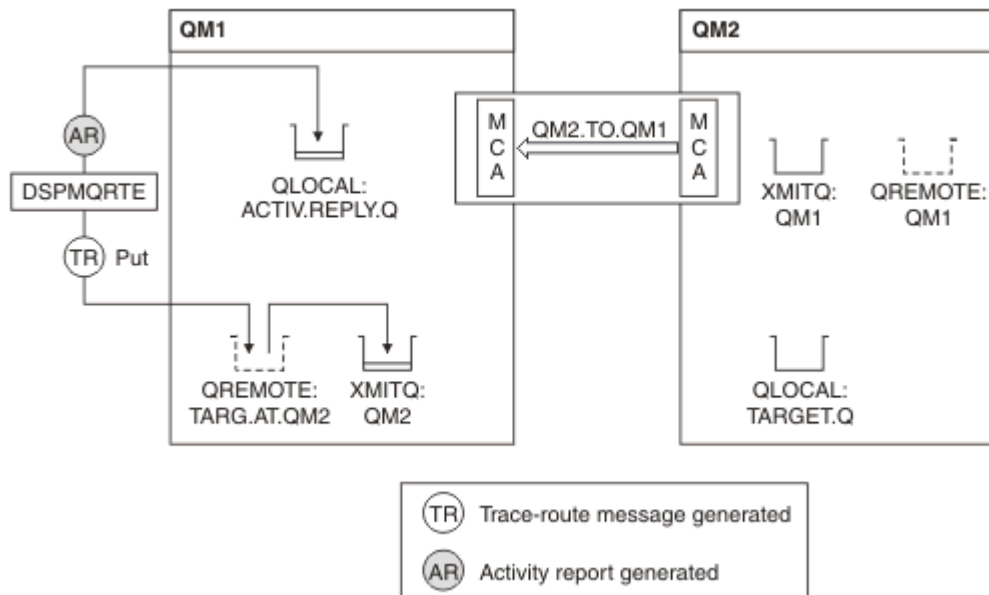


Figura 19. Diagnosticar un problema de canal

- En este ejemplo, el canal QM1.TO.QM2 no está en ejecución.
- DSPMQRTE coloca un mensaje de ruta de rastreo (como en el ejemplo 1) en la cola de destino, y genera un informe de actividad.
- No hay ningún MCA para obtener el mensaje de la cola de transmisión (QM2), por lo tanto, este es el único informe de actividad que DSPMQRTE recupera de la cola de respuesta. Esta vez, el hecho de que la ruta no se haya completado indica que hay un problema. El administrador puede utilizar la cola de transmisión encontrada en ResolvedQName para investigar por qué no se da servicio a la cola de transmisión.

La salida que se muestra es la siguiente:

```

AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
        -rq ACTIV.REPLY.Q -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
        queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'

  Operation:
    OperationType: Put
    QMgrName: 'QM1'
    QName: 'TARG.AT.QM2'
    ResolvedQName: 'QM2'
    RemoteQName: 'TARGET.Q'
    RemoteQMGrName: 'QM2'
-----
AMQ8652: DSPMQRTE command has finished.

```

Referencia de informe de actividad

Utilice esta página para obtener una visión general del formato de mensaje de informe de actividad. Los datos de mensaje de informe de actividad contienen los parámetros que describen la actividad.

Formato de informe de actividad

Los informes de actividad son mensajes de informe estándar de IBM MQ que contienen un descriptor de mensaje y datos de mensaje. Los informes de actividad son mensajes PCF generados por las aplicaciones

que han realizado una actividad en nombre de un mensaje mientras se ha direccionado a través de una red de gestores de colas.

Los informes de actividad contienen la siguiente información;

Un descriptor de mensaje

Una estructura MQMD

Datos de mensaje

Constan de lo siguiente:

- Una cabecera PCF incrustada (MQEPH).
- Datos del mensaje de informe de actividad.

Los datos de mensaje de informe de actividad constan del grupo PCF *Activity* y, si se ha generado para un mensaje de ruta de rastreo, el grupo PCF *TraceRoute*.

En la [Tabla 19](#) en la [página 102](#) se muestra la estructura de estos informes, incluidos los parámetros que se devuelven sólo bajo ciertas condiciones.

Tabla 19. Formato de informe de actividad

estructura MQMD	Estructura MQEPH de cabecera PCF integrada	Datos de mensaje de informe de actividad
Identificador de la estructura Versión de la estructura Opciones de informe Tipo de mensaje Hora de caducidad Comentarios Encoding ID de juego de caracteres codificado Formato de mensaje Prioridad Persistence Identificador de mensaje Identificador de correlación Recuento de restituciones Cola de respuestas Gestor de colas de respuestas Identificador de usuario Señal de contabilidad Datos de identidad de la aplicación Tipo de aplicación Nombre de la aplicación Fecha de transferencia Hora de transferencia Datos de origen de la aplicación Identificador de grupo Número de secuencia de mensajes Desplazamiento Distintivos de mensajes Longitud original	Identificador de la estructura Versión de la estructura Longitud de estructura Encoding ID de juego de caracteres codificado Formato de mensaje Flags Cabecera PCF (MQCFH) Tipo de estructura Longitud de estructura Versión de la estructura Identificador de mandato Número de secuencia de mensajes Opciones de control Código de terminación Código de razón Recuento de parámetros	Actividad Nombre de la aplicación de la actividad Tipo de aplicación de la actividad Descripción de la actividad Operación Tipo de operación Fecha de operación Hora de operación Mensaje Longitud del mensaje MQMD ⁸ EmbeddedMQMD Nombre del gestor de colas Nombre del grupo de compartición de colas Nombre de cola ¹ ² ³ ⁷ Nombre de cola resuelto ¹ ³ ⁷ Nombre de cola remota ³ ⁷ Nombre de gestor de colas remoto ² ³ ⁴ ⁵ ⁷ Nivel de suscripción ⁹ Identificador de suscripción ⁹ Retroalimentación ² ¹⁰ Nombre de canal ⁴ ⁵ Tipo de canal ⁴ ⁵ Nombre de cola de transmisión ⁵ TraceRoute ⁶ Detail Actividades registradas Actividades no registradas Recuento de discontinuidad Núm. máximo de actividades Accumulate Deliver

Notas:

1. Se devuelve para las operaciones get y browse.
2. Se devuelve para las operaciones Discard.
3. Se devuelve para las operaciones Put, Put Reply y Put Report.
4. Se devuelve para las operaciones Receive.

5. Se devuelve para las operaciones Send.
6. Se devuelve para los mensajes de ruta de rastreo.
7. No se devuelve para las operaciones put efectuadas hacia un tema, contenido dentro de las actividades de Publish.
8. No se devuelve para las operaciones Excluded Publish. Para las operaciones Publish y Discarded Publish, se devuelve, conteniendo un subconjunto de parámetros.
9. Se devuelve para las operaciones Publish, Discarded Publish y Excluded Publish.
10. Se devuelve para las operaciones Discarded Publish y Excluded Publish.

MQMD de informe de actividad (descriptor de mensaje)

Utilice esta página para ver los valores contenidos en la estructura MQMD de un informe de actividad.

StrucId

Identificador de estructura:

Tipo de datos

MQCHAR4

Valor

MQMD_STRUC_ID.

Version

Número de versión de la estructura

Tipo de datos

MQLONG

Valores

Se copia del descriptor de mensaje original. Los valores posibles son:

MQMD_VERSION_1

Estructura de descriptor de mensaje de Versión-1, soportada en todos los entornos.

MQMD_VERSION_2

Estructura de descriptor de mensaje de versión 2, soportada en los entornos siguientes:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

y todos los IBM MQ MQI clients conectados a estos sistemas.

Report

Opciones para más mensajes de informe

Tipo de datos

MQLONG

Valor

Si se ha especificado MQRO_PASS_DISCARD_AND_EXPIRY o MQRO_DISCARD_MSG en el campo *Informe* del descriptor de mensaje original:

MQRO_DISCARD

Se descarta el informe si no se puede entregar a la cola de destino.

De lo contrario:

MQRO_NONE

No se necesita ningún informe.

MsgType

Indica el tipo de mensaje.

Tipo de datos

MQLONG

Valor

MQMT_REPORT

Expiry

Duración del mensaje del informe

Tipo de datos

MQLONG

Valor

Si el campo *Informe* del descriptor de mensaje original se ha especificado como MQRO_PASS_DISCARD_AND_EXPIRY, se utiliza el tiempo de caducidad restante del mensaje original.

De lo contrario:

MQEI_UNLIMITED

El informe no tiene tiempo de caducidad.

Feedback

Descripción: Código de retorno o de razón.

Tipo de datos: MQLONG.

Valor: **MQFB_ACTIVITY**
Informe de actividad.

Encoding

Descripción: Codificación numérica de los datos de mensaje del informe.

Tipo de datos: MQLONG.

Valor: MQENC_NATIVE.

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensaje del informe.

Tipo de datos: MQLONG.

Valor: Establézcase según convenga.

Format

Descripción: Nombre de formato de los datos de mensaje del informe

Tipo de datos: MQCHAR8.

Valor: **MQFMT_EMBEDDED_PCF**
Mensajes PCF integrados.

Priority

Descripción: Prioridad del mensaje de informe.

Tipo de datos: MQLONG.

Valor: Se copia del descriptor de mensaje original.

Persistence

Descripción: Persistencia del mensaje del informe.
Tipo de datos: MQLONG.
Valor: Se copia del descriptor de mensaje original.

MsgId

Descripción: Identificador del mensaje.
Tipo de datos: MQBYTE24.
Valores: Si el campo *Informe* del descriptor de mensaje original se ha especificado como MQRO_PASS_MSG_ID, se utiliza el identificador de mensaje del mensaje original.
De lo contrario, el gestor de colas generará un valor exclusivo.

CorrelId

Descripción: Identificador de correlación.
Tipo de datos: MQBYTE24.
Valor: Si el campo *Informe* del descriptor de mensaje original se ha especificado como MQRO_PASS_CORREL_ID, se utiliza el identificador de correlación del mensaje original.
De lo contrario, el identificador de mensaje se copia del mensaje original.

BackoutCount

Descripción: Umbral de restitución.
Tipo de datos: MQLONG.
Valor: 0.

ReplyToQ

Descripción: Nombre de la cola de respuesta.
Tipo de datos: MQCHAR48.
Valores: En blanco.

ReplyToQMGr

Descripción: Nombre del gestor de colas de respuesta.
Tipo de datos: MQCHAR48.
Valor: El nombre del gestor de colas que ha generado el mensaje de informe.

UserIdentifier

Descripción: El identificador de usuario de la aplicación que ha generado el mensaje de informe.
Tipo de datos: MQCHAR12.
Valor: Se copia del descriptor de mensaje original.

AccountingToken

Descripción:	Señal de contabilidad que permite que una aplicación cargo el trabajo realizado como resultado del mensaje.
Tipo de datos:	MQBYTE32.
Valor:	Se copia del descriptor de mensaje original.

ApplIdentityData

Descripción:	Datos de la aplicación relacionados con la identidad.
Tipo de datos:	MQCHAR32.
Valores:	Se copia del descriptor de mensaje original.

PutApplType

Descripción:	Tipo de aplicación que ha transferido el mensaje de informe.
Tipo de datos:	MQLONG.
Valor:	MQAT_QMGR Mensaje generado por el gestor de colas.

PutApplName

Descripción:	Nombre de la aplicación que transfiere el mensaje de informe.
Tipo de datos:	MQCHAR28.
Valor:	Bien los primeros 28 bytes del nombre del gestor de colas, o bien el nombre del MCA que ha generado el mensaje de informe.

PutDate

Descripción:	Fecha en que se transfirió el mensaje.
Tipo de datos:	MQCHAR8.
Valor:	Tal como lo genere el gestor de colas.

PutTime

Descripción:	Hora en que se transfirió el mensaje.
Tipo de datos:	MQCHAR8.
Valor:	Tal como lo genere el gestor de colas.

ApplOriginData

Descripción:	Datos de la aplicación relacionados con el origen.
Tipo de datos:	MQCHAR4.
Valor:	En blanco.

Si *Version* es MQMD_VERSION_2, están presentes los campos adicionales siguientes:

GroupId

Descripción:	Identifica a qué grupo de mensajes o mensaje lógico pertenece el mensaje físico.
Tipo de datos:	MQBYTE24.

Valor: Se copia del descriptor de mensaje original.

MsgSeqNumber

Descripción: Número de secuencia del mensaje lógico en el grupo.

Tipo de datos: MQLONG.

Valor: Se copia del descriptor de mensaje original.

Offset

Descripción: Desplazamiento de datos en el mensaje físico desde el inicio del mensaje lógico.

Tipo de datos: MQLONG.

Valor: Se copia del descriptor de mensaje original.

MsgFlags

Descripción: Distintivos de mensaje que especifican los atributos del mensaje, o controlan su proceso.

Tipo de datos: MQLONG.

Valor: Se copia del descriptor de mensaje original.

OriginalLength

Descripción: Longitud del mensaje original.

Tipo de datos: MQLONG.

Valor: Se copia del descriptor de mensaje original.

MQEPH de informe de actividad (cabecera PCF incorporada)

Utilice esta página para ver los valores contenidos en la estructura MQEPH de un informe de actividad.

La estructura MQEPH contiene una descripción tanto de la información PCF que acompaña a los datos de mensaje de un informe de actividad, como de los datos de mensaje de aplicación que le siguen.

Para un informe de actividad, la estructura MQEPH contiene los valores siguientes:

StrucId

Descripción: Identificador de estructura.

Tipo de datos: MQCHAR4.

Valor: MQEPH_STRUC_ID.

Version

Descripción: Número de versión de la estructura.

Tipo de datos: MQLONG.

Valores: MQEPH_VERSION_1.

StrucLength

Descripción: Longitud de la estructura.

Tipo de datos: MQLONG.

Valor: Longitud total de la estructura, incluidas las estructuras de parámetros PCF que le siguen.

Encoding

Descripción:	Codificación numérica de los datos de mensaje que siguen a la última estructura de parámetros PCF.
Tipo de datos:	MQLONG.
Valor:	Si alguno de los datos del mensaje de aplicación original se incluye en el mensaje de informe, el valor se copiará del campo <i>Encoding</i> del descriptor de mensaje original. En caso contrario, 0.

CodedCharSetId

Descripción:	Identificador de juego de caracteres de los datos de mensaje que siguen a la última estructura de parámetros PCF.
Tipo de datos:	MQLONG.
Valor:	Si alguno de los datos del mensaje de aplicación original se incluye en el mensaje de informe, el valor se copiará del campo <i>CodedCharSetId</i> del descriptor de mensaje original. En caso contrario, MQCCSI_UNDEFINED.

Format

Descripción:	Nombre del formato de los datos de mensaje que siguen a la última estructura de parámetros PCF.
Tipo de datos:	MQCHAR8.
Valor:	Si alguno de los datos del mensaje de aplicación original se incluye en el mensaje de informe, el valor se copiará del campo <i>Format</i> del descriptor de mensaje original. En caso contrario, MQFMT_NONE.

Flags

Descripción:	Distintivos que especifican los atributos de la estructura, o controlan su proceso.
Tipo de datos:	MQLONG.
Valor:	MQEPH_CCSID_EMBEDDED Especifica que el conjunto de caracteres de los parámetros que contienen los datos de caracteres se especifica individualmente, dentro del campo <i>CodedCharSetId</i> , de cada estructura.

PCFHeader

Descripción:	Cabecera de formato de mandato programable
Tipo de datos:	MQCFH.
Valor:	Consulte “MQCFH de informe de actividad (cabecera PCF)” en la página 108.

MQCFH de informe de actividad (cabecera PCF)

Utilice esta página para ver los valores PCF contenidos en la estructura MQCFH de un informe de actividad.

Para un informe de actividad, la estructura MQCFH contiene los valores siguientes:

Type

Descripción: Tipo de estructura que identifica el contenido del mensaje de informe.
Tipo de datos: MQLONG.
Valor: **MQCFT_REPORT**
El mensaje es un informe.

StrucLength

Descripción: Longitud de la estructura.
Tipo de datos: MQLONG.
Valor: **MQCFH_STRUC_LENGTH**
Longitud en bytes de la estructura MQCFH.

Version

Descripción: Número de versión de la estructura.
Tipo de datos: MQLONG.
Valores: MQCFH_VERSION_3

Command

Descripción: Identificador de mandato. Identifica la categoría del mensaje.
Tipo de datos: MQLONG.
Valores: **MQCMD_ACTIVITY_MSG**
Actividad de mensaje.

MsgSeqNumber

Descripción: Número de secuencia de mensaje. Éste es el número de secuencia del mensaje dentro de un grupo de mensajes relacionados.
Tipo de datos: MQLONG.
Valores: 1.

Control

Descripción: Opciones de control.
Tipo de datos: MQLONG.
Valores: MQCFC_LAST.

CompCode

Descripción: Código de terminación.
Tipo de datos: MQLONG.
Valores: MQCC_OK.

Reason

Descripción: Código de razón que califica el código de terminación.
Tipo de datos: MQLONG.

Valores: MQRC_NONE.

ParameterCount

Descripción: Recuento de las estructuras de parámetro. Es el número de estructuras de parámetro que siguen la estructura MQCFH. Una estructura de grupo (MQCFGR) y las estructuras de parámetro que incluye, se consideran como una estructura única.

Tipo de datos: MQLONG.

Valores: 1 o mayor.

Datos de mensaje de informe de actividad

Utilice esta página para ver los parámetros el grupo PCF *Activity* contiene en un mensaje de informe de actividad. Algunos parámetros sólo se devuelven cuando se han realizado operaciones específicas.

Los datos de mensaje de informe de actividad constan del grupo PCF *Activity* y, si se ha generado para un mensaje de ruta de rastreo, el grupo PCF *TraceRoute*. En este tema se describe el grupo PCF *Activity* de forma detallada.

Algunos parámetros, que se describen como Datos de mensaje de informe de actividad específicos de operación, sólo se devuelven cuando se han realizado operaciones específicas.

En un informe de actividad, los datos de mensaje de informe de actividad contienen los parámetros siguientes:

Activity

Descripción: Parámetros agrupados que describen la actividad.

Identificador: MQGACF_ACTIVITY.

Tipo de datos: MQCFGR.

Incluido en el grupo PCF: Ninguna.

Parámetros del grupo PCF: *ActivityApplName*
ActivityApplType
ActivityDescription
Operation
TraceRoute

Se devuelve: Siempre.

ActivityApplName

Descripción: Nombre de la aplicación que ha realizado la actividad.

Identificador: MQCACF_APPL_NAME.

Tipo de datos: MQCFST.

Incluido en el grupo PCF: *Activity*.

Longitud máxima: MQ_APPL_NAME_LENGTH.

Se devuelve: Siempre.

ActivityApplType

Descripción: Tipo de aplicación que ha realizado la actividad.

Identificador: MQIA_APPL_TYPE.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: *Activity*.
Se devuelve: Siempre.

ActivityDescription

Descripción: Descripción de la actividad que realiza la aplicación.
Identificador: MQCACF_ACTIVITY_DESCRIPTION.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: *Activity*.
Longitud máxima: 64
Se devuelve: Siempre.

Operation

Descripción: Parámetros agrupados que describen una operación de la actividad.
Identificador: MQGACF_OPERATION.
Tipo de datos: MQCFGR.
Incluido en el grupo PCF: *Activity*.
Parámetros del grupo PCF: *OperationType*
OperationDate
OperationTime
Message
QMgrName
QSGName

Nota: En este grupo se devuelven parámetros adicionales, según el tipo de operación. Estos parámetros adicionales se describen como Datos de mensaje de informe de actividad específicos de operación.

Se devuelve: Un grupo PCF *Operation* por operación en la actividad.

OperationType

Descripción: Tipo de operación que se realiza.
Identificador: MQIACF_OPERATION_TYPE.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: *Operation*.
Valores: MQOPER_*.
Se devuelve: Siempre.

OperationDate

Descripción: Fecha en que se ha realizado la operación.

Identificador: MQCACF_OPERATION_DATE.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: *Operation.*
Longitud máxima: MQ_DATE_LENGTH.
Se devuelve: Siempre.

OperationTime

Descripción: Hora en que se ha realizado la operación.
Identificador: MQCACF_OPERATION_TIME.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: *Operation.*
Longitud máxima: MQ_TIME_LENGTH.
Se devuelve: Siempre.

Message

Descripción: Parámetros agrupados que describen el mensaje que ha provocado la actividad.
Identificador: MQGACF_MESSAGE.
Tipo de datos: MQCFGR.
Incluido en el grupo PCF: *Operation.*
Parámetros del grupo: *MsgLength*
MQMD
EmbeddedMQMD
Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

MsgLength

Descripción: Longitud del mensaje que ha provocado la actividad, antes de que se haya producido la actividad.
Identificador: MQIACF_MSG_LENGTH.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: *Message.*
Se devuelve: Siempre.

MQMD

Descripción: Parámetros agrupados relacionados con el descriptor de mensaje del mensaje que ha provocado la actividad.
Identificador: MQGACF_MQMD.
Tipo de datos: MQCFGR.
Incluido en el grupo PCF: *Message.*

Parámetros del grupo:	<i>StrucId</i> <i>Version</i> <i>Report</i> <i>MsgType</i> <i>Expiry</i> <i>Feedback</i> <i>Encoding</i> <i>CodedCharSetId</i> <i>Format</i> <i>Priority</i> <i>Persistence</i> <i>MsgId</i> <i>CorrelId</i> <i>BackoutCount</i> <i>ReplyToQ</i> <i>ReplyToQMgr</i> <i>UserIdentifier</i> <i>AccountingToken</i> <i>ApplIdentityData</i> <i>PutApplType</i> <i>PutApplName</i> <i>PutDate</i> <i>PutTime</i> <i>ApplOriginData</i> <i>GroupId</i> <i>MsgSeqNumber</i> <i>Offset</i> <i>MsgFlags</i> <i>OriginalLength</i>
-----------------------	---

Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

EmbeddedMQMD

Descripción:	Parámetros agrupados que describen el descriptor de mensaje incrustado dentro de un mensaje en una cola de transmisión.
Identificador:	MQGACF_EMBEDDED_MQMD.
Tipo de datos:	MQCFGR.
Incluido en el grupo PCF:	<i>Message</i> .

Parámetros del grupo:

StrucId
Version
Report
MsgType
Expiry
Feedback
Encoding
CodedCharSetId
Format
Priority
Persistence
MsgId
CorrelId
BackoutCount
ReplyToQ
ReplyToQMgr
UserIdentifier
AccountingToken
ApplIdentityData
PutApplType
PutApplName
PutDate
PutTime
ApplOriginData
GroupId
MsgSeqNumber
Offset
MsgFlags
OriginalLength

Se devuelve: Para las operaciones Get en las que la cola se resuelve en una cola de transmisión.

StrucId

Descripción: Identificador de la estructura

Identificador: MQCACF_STRUC_ID.

Tipo de datos: MQCFST.

Incluido en el grupo PCF: MQMD o *EmbeddedMQMD*.

Longitud máxima: 4.

Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

Version

Descripción: Número de versión de la estructura.

Identificador: MQIACF_VERSION.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*

Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

Report

Descripción: Opciones para los mensajes de informe.

Identificador: MQIACF_REPORT.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*

Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

MsgType

Descripción: Indica el tipo de mensaje.

Identificador: MQIACF_MSG_TYPE.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*

Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

Expiry

Descripción: Duración del mensaje.

Identificador: MQIACF_EXPIRY.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*

Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

Feedback

Descripción: Código de retorno o de razón.

Identificador: MQIACF_FEEDBACK.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*

Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

Encoding

Descripción: Codificación numérica de datos de mensaje.

Identificador: MQIACF_ENCODING.

Tipo de datos: MQCFIN.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensajes.
Identificador: MQIA_CODED_CHAR_SET_ID.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

Format

Descripción: Nombre de formato de datos de mensaje
Identificador: MQCACH_FORMAT_NAME.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Longitud máxima: MQ_FORMAT_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

Priority

Descripción: Prioridad del mensaje.
Identificador: MQIACF_PRIORITY.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

Persistence

Descripción: Persistencia de mensajes.
Identificador: MQIACF_PERSISTENCE.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

MsgId

Descripción: Identificador del mensaje.
Identificador: MQBACF_MSG_ID.

Tipo de datos: MQCFBS.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Longitud máxima: MQ_MSG_ID_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

CorrelId

Descripción: Identificador de correlación.
Identificador: MQBACF_CORREL_ID.
Tipo de datos: MQCFBS.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Longitud máxima: MQ_CORREL_ID_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish.

BackoutCount

Descripción: Umbral de restitución.
Identificador: MQIACF_BACKOUT_COUNT.
Tipo de datos: MQCFIN.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

ReplyToQ

Descripción: Nombre de la cola de respuesta.
Identificador: MQCACF_REPLY_TO_Q.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Longitud máxima: MQ_Q_NAME_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

ReplyToQMgr

Descripción: Nombre del gestor de colas de respuesta.
Identificador: MQCACF_REPLY_TO_Q_MGR.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: MQMD o EmbeddedMQMD.
Longitud máxima: MQ_Q_MGR_NAME_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

UserIdentifier

Descripción:	El identificador de usuario de la aplicación que ha originado el mensaje.
Identificador:	MQCACF_USER_IDENTIFIER.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Longitud máxima:	MQ_USER_ID_LENGTH.
Se devuelve:	Siempre, excepto para las operaciones Excluded Publish.

AccountingToken

Descripción:	Señal de contabilidad que permite que una aplicación cargo el trabajo realizado como resultado del mensaje.
Identificador:	MQBACF_ACCOUNTING_TOKEN.
Tipo de datos:	MQCFBS.
Incluido en el grupo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Longitud máxima:	MQ_ACCOUNTING_TOKEN_LENGTH.
Se devuelve:	Siempre, excepto para las operaciones Excluded Publish.

ApplIdentityData

Descripción:	Datos de la aplicación relacionados con la identidad.
Identificador:	MQCACF_APPL_IDENTITY_DATA.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Longitud máxima:	MQ_APPL_IDENTITY_DATA_LENGTH.
Se devuelve:	Siempre, excepto para las operaciones Excluded Publish.

PutApplType

Descripción:	Tipo de aplicación que ha transferido el mensaje.
Identificador:	MQIA_APPL_TYPE.
Tipo de datos:	MQCFIN.
Incluido en el grupo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Se devuelve:	Siempre, excepto para las operaciones Excluded Publish y, en <i>MQMD</i> , para las operaciones Publish y Discarded Publish.

PutApplName

Descripción:	Nombre de la aplicación que transfiere el mensaje.
Identificador:	MQCACF_APPL_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .

Longitud máxima: MQ_APPL_NAME_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

PutDate

Descripción: Fecha en que se transfirió el mensaje.
Identificador: MQCACF_PUT_DATE.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*
Longitud máxima: MQ_PUT_DATE_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

PutTime

Descripción: Hora en que se transfirió el mensaje.
Identificador: MQCACF_PUT_TIME.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*
Longitud máxima: MQ_PUT_TIME_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

ApplOriginData

Descripción: Datos de la aplicación relacionados con el origen.
Identificador: MQCACF_APPL_ORIGIN_DATA.
Tipo de datos: MQCFST.
Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*
Longitud máxima: MQ_APPL_ORIGIN_DATA_LENGTH.
Se devuelve: Siempre, excepto para las operaciones Excluded Publish y, en MQMD, para las operaciones Publish y Discarded Publish.

GroupId

Descripción: Identifica a qué grupo de mensajes o mensaje lógico pertenece el mensaje físico.
Identificador: MQBACF_GROUP_ID.
Tipo de datos: MQCFBS.
Incluido en el grupo PCF: *MQMD o EmbeddedMQMD.*
Longitud máxima: MQ_GROUP_ID_LENGTH.

Se devuelve: Si *Version* se especifica como MQMD_VERSION_2. No se devuelve en operaciones Excluded Publish y, en MQMD, para la operaciones Publish y Discarded Publish.

MsgSeqNumber

Descripción: Número de secuencia del mensaje lógico en el grupo.

Identificador: MQIACH_MSG_SEQUENCE_NUMBER.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: MQMD o EmbeddedMQMD.

Se devuelve: Si *Version* se especifica como MQMD_VERSION_2. No se devuelve en operaciones Excluded Publish y, en MQMD, para la operaciones Publish y Discarded Publish.

Offset

Descripción: Desplazamiento de datos en el mensaje físico desde el inicio del mensaje lógico.

Identificador: MQIACF_OFFSET.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: MQMD o EmbeddedMQMD.

Se devuelve: Si *Version* se especifica como MQMD_VERSION_2. No se devuelve en operaciones Excluded Publish y, en MQMD, para la operaciones Publish y Discarded Publish.

MsgFlags

Descripción: Distintivos de mensaje que especifican los atributos del mensaje, o controlan su proceso.

Identificador: MQIACF_MSG_FLAGS.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: MQMD o EmbeddedMQMD.

Se devuelve: Si *Version* se especifica como MQMD_VERSION_2. No se devuelve en operaciones Excluded Publish y, en MQMD, para la operaciones Publish y Discarded Publish.

OriginalLength

Descripción: Longitud del mensaje original.

Identificador: MQIACF_ORIGINAL_LENGTH.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: MQMD o EmbeddedMQMD.

Se devuelve: Si *Version* se especifica como MQMD_VERSION_2. No se devuelve en operaciones Excluded Publish y, en MQMD, para la operaciones Publish y Discarded Publish.

QMgrName

Descripción:	Nombre del gestor de colas donde se realizó la actividad.
Identificador:	MQCA_Q_MGR_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH
Se devuelve:	Siempre.

QSGName

Descripción:	Nombre del grupo de compartición de colas al que pertenece el gestor de colas donde se realizó la actividad.
Identificador:	MQCA_QSG_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_QSG_NAME_LENGTH
Se devuelve:	Si la actividad se ha realizado en un gestor de colas de IBM MQ for z/OS.

TraceRoute

Descripción:	Parámetros agrupados que especifican los atributos del mensaje de ruta de rastreo.
Identificador:	MQGACF_TRACE_ROUTE.
Tipo de datos:	MQCFGR.
Contenidos en el grupo PCF:	<i>Activity.</i>
Parámetros del grupo:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
Se devuelve:	Si la actividad se ha realizado en nombre del mensaje de ruta de rastreo.

Los valores de los parámetros del grupo PCF *TraceRoute* son los del mensaje de ruta de rastreo en el momento en que se haya generado el informe de actividad.

Datos de mensaje de informe de actividad específicos de operación

Utilice esta página para ver los parámetros PCF adicionales que se podrían devolver en los grupo PCF *Operation* en un informe de actividad, dependiendo del valor del parámetro *OperationType*.

Los parámetros adicionales varían en función de los tipos de operación siguientes:

Get/Browse (MQOPER_GET/MQOPER_BROWSE)

Los parámetros de datos de mensajes de informe de actividad adicional que se devuelven en el valor *Operation* del grupo PCF, para el tipo de operación Get/Browse (MQOPER_GET/MQOPER_BROWSE) (se ha obtenido, o se ha examinado, un mensaje en una cola).

QName

Descripción:	El nombre de la cola que se ha abierto.
Identificador:	MQCA_Q_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_Q_NAME_LENGTH
Se devuelve:	Siempre.

ResolvedQName

Descripción:	El nombre en que resuelve la cola abierta.
Identificador:	MQCACF_RESOLVED_Q_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_Q_NAME_LENGTH
Se devuelve:	Siempre.

Discard (MQOPER_DISCARD)

Los parámetros de datos de mensajes de informe de actividad adicional que se devuelven en el valor *Operation* del grupo PCF, para el tipo de operación Discard (MQOPER_DISCARD) (se ha rechazado un mensaje).

Feedback

Descripción:	La razón por la que se ha rechazado el mensaje.
Identificador:	MQIACF_FEEDBACK.
Tipo de datos:	MQCFIN.
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Siempre.

QName

Descripción:	El nombre de la cola que se ha abierto.
Identificador:	MQCA_Q_NAME.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_Q_NAME_LENGTH
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Si el mensaje se ha descartado porque se transfirió de forma no satisfactoria a una cola.

RemoteQMgrName

Descripción:	El nombre del gestor de colas que era el destino del mensaje.
Identificador:	MQCA_REMOTE_Q_MGR_NAME.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Si el valor de <i>Feedback</i> es MQFB_NOT_FORWARDED.

Publish/Discarded Publish/Excluded Publish (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)

Los parámetros de datos de mensajes de informe de actividad adicional que se devuelven en el grupo PCF *Operation* para el tipo de operación *Publish/Discarded Publish/Excluded Publish (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)* (se ha entregado, descartado o excluido un mensaje de publicación/suscripción).

SubId

Descripción:	El identificador de suscripción.
Identificador:	MQBACF_SUB_ID.
Tipo de datos:	MQCFBS.
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Siempre.

SubLevel

Descripción:	El nivel de suscripción.
Identificador:	MQIACF_SUB_LEVEL.
Tipo de datos:	MQCFIN.
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Siempre.

Feedback

Descripción:	La razón por la que se descarta el mensaje.
Identificador:	MQIACF_FEEDBACK.
Tipo de datos:	MQCFIN.
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Si el mensaje se ha descartado porque no se ha entregado a un suscriptor, o el mensaje no se ha entregado porque se ha excluido el suscriptor.

La operación *Publish*, *MQOPER_PUBLISH*, proporciona información sobre un mensaje entregado a un suscriptor en particular. En esta operación se describen los elementos del mensaje posterior que pueden haber cambiado a partir del mensaje descrito en la operación *Put* asociada. De forma similar a una operación *Put*, contiene un grupo de mensajes *MQGACF_MESSAGE* y, dentro de éste, un grupo *MQMD MQGACF_MQMD*. No obstante, este grupo *MQMD* contiene solo los campos siguientes, que un

suscriptor puede alterar temporalmente: *Format, Priority, Persistence, MsgId, CorrelId, UserIdentifier, AccountingToken, ApplIdentityData*.

Los valores *SubId* y *SubLevel* del suscriptor se incluyen en la información de la operación. Puede utilizar el valor *SubID* con el mandato PCF MQCMD_INQUIRE_SUBSCRIBER para recuperar todos los demás atributos de un suscriptor.

La operación Discarded Publish, MQOPER_DISCARDED_PUBLISH, es análoga a la operación Discard que se utiliza cuando un mensaje no se entrega en la mensajería punto a punto. Un mensaje no se entrega a un suscriptor si se ha solicitado explícitamente que el mensaje no se entregue en un destino local, y este suscriptor especifica un destino local. También se considera que no mensaje no se ha entregado si existe un problema para hacer llegar el mensaje a la cola de destino, por ejemplo, porque la cola está llena.

La información de una operación Discarded Publish es la misma que para una operación Publish, con la adición de un campo *Feedback* que ofrece las razones por las que no se ha entregado el mensaje. Este campo *Feedback* contiene los valores MQFB_* o MQRC_*, que son comunes con la operación MQOPER_DISCARD. La razón para descartar una publicación, en contraposición a excluirla, son las mismas que las razones por las que se descarta una operación de tipo Put.

La operación Excluded Publish, MQOPER_EXCLUDED_PUBLISH, proporciona información sobre un suscriptor que se tenía en cuenta para la entrega del mensaje, ya que el tema en el que el suscriptor realizaba la suscripción coincide con el de la operación Put asociada, pero el mensaje no se ha entregado al suscriptor debido a que existen otros criterios de selección no coinciden con el mensaje que se transfiere en el tema. Al igual que ocurre con una operación Discarded Publish, el campo *Feedback* proporciona información sobre el motivo por el cual se ha excluido esta suscripción. No obstante, a diferencia de la operación Discarded Publish, no se proporciona ninguna información relacionada con mensajes porque no se ha generado ningún mensaje para dicho suscriptor.

Put/Put Reply/Put Report (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT)

Los parámetros de datos de mensajes de informe de actividad adicional que se devuelven en el valor *Operation* del grupo PCF, para el tipo de operación Put/Put Reply/Put Report (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT) (se ha transferido a una cola un mensaje, un mensaje de respuesta o mensaje de informe).

QName

Descripción:	El nombre de la cola que se ha abierto.
Identificador:	MQCA_Q_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation</i> .
Longitud máxima:	MQ_Q_NAME_LENGTH
Se devuelve:	Siempre, aparte de una excepción: no se devuelve si la operación put es hacia un tema, contenido dentro de una actividad de publicación.

ResolvedQName

Descripción:	El nombre en que resuelve la cola abierta.
Identificador:	MQCAF_RESOLVED_Q_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation</i> .
Longitud máxima:	MQ_Q_NAME_LENGTH

Se devuelve: Cuando la cola abierta se ha podido resolver. No se devuelve si la operación put es hacia un tema, contenido dentro de una actividad de publicación.

RemoteQName

Descripción: El nombre de la cola que se abre, tal como se la conoce en el gestor de colas remoto.

Identificador: MQCA_REMOTE_Q_NAME.

Tipo de datos: MQCFST.

Incluido en el grupo PCF: *Operation.*

Longitud máxima: MQ_Q_NAME_LENGTH

Se devuelve: Si la cola abierta es una cola remota. No se devuelve si la operación put es hacia un tema, contenido dentro de una actividad de publicación.

RemoteQMgrName

Descripción: Nombre del gestor de colas remoto en el que se ha definido la cola remota.

Identificador: MQCA_REMOTE_Q_MGR_NAME.

Tipo de datos: MQCFST.

Incluido en el grupo PCF: *Operation.*

Longitud máxima: MQ_Q_MGR_NAME_LENGTH

Se devuelve: Si la cola abierta es una cola remota. No se devuelve si la operación put es hacia un tema, contenido dentro de una actividad de publicación.

TopicString

Descripción: La serie de tema completa a la que se transfiere el mensaje.

Identificador: MQCA_TOPIC_STRING.

Tipo de datos: MQCFST.

Incluido en el grupo PCF: *Operation.*

Se devuelve: Si la operación put se realiza hacia un tema, contenido dentro de una actividad de publicación.

Feedback

Descripción: La razón del mensaje que se transfiere a la cola de mensajes no entregados.

Identificador: MQIACF_FEEDBACK.

Tipo de datos: MQCFIN.

Incluido en el grupo PCF: *Operation.*

Se devuelve: Si el mensaje se transfirió a la cola de mensajes no entregados.

Receive (MQOPER_RECEIVE)

Los parámetros de datos de mensajes de informe de actividad adicional que se devuelven en el valor *Operation* del grupo PCF, para el tipo de operación Receive (MQOPER_RECEIVE) (se ha recibido un mensaje en un canal).

ChannelName

Descripción:	El nombre del canal en el que se ha recibido el mensaje.
Identificador:	MQCACH_CHANNEL_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_CHANNEL_NAME_LENGTH
Se devuelve:	Siempre.

ChannelType

Descripción:	El tipo de canal en el que se ha recibido el mensaje.
Identificador:	MQIACH_CHANNEL_TYPE.
Tipo de datos:	MQCFIN.
Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Siempre.

RemoteQMgrName

Descripción:	El nombre del gestor de colas del cual se ha recibido el mensaje.
Identificador:	MQCA_REMOTE_Q_MGR_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH
Se devuelve:	Siempre.

Send (MQOPER_SEND)

Los parámetros de datos de mensajes de informe de actividad adicional que se devuelven en el valor *Operation* del grupo PCF, para el tipo de operación Send (MQOPER_SEND) (se ha enviado un mensaje en un canal).

ChannelName

Descripción:	El nombre del canal donde se envió el mensaje.
Identificador:	MQCACH_CHANNEL_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_CHANNEL_NAME_LENGTH.
Se devuelve:	Siempre.

ChannelType

Descripción:	El tipo de canal donde se envió el mensaje.
Identificador:	MQIACH_CHANNEL_TYPE.
Tipo de datos:	MQCFIN.

Incluido en el grupo PCF:	<i>Operation.</i>
Se devuelve:	Siempre.

XmitQName

Descripción:	La cola de transmisión de la que se ha recuperado el mensaje.
Identificador:	MQCACH_XMIT_Q_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_Q_NAME_LENGTH.
Se devuelve:	Siempre.

RemoteQMgrName

Descripción:	El nombre del gestor de colas remoto al que se envió el mensaje.
Identificador:	MQCA_REMOTE_Q_MGR_NAME.
Tipo de datos:	MQCFST.
Incluido en el grupo PCF:	<i>Operation.</i>
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH
Se devuelve:	Siempre.

Referencia de mensaje de ruta de rastreo

Utilice esta página para obtener una visión general del formato de mensaje de ruta de rastreo. Los datos del mensaje de ruta de rastreo incluyen parámetros que describen las actividades que el mensaje de ruta de rastreo ha provocado.

Formato de mensaje de ruta de rastreo

Los mensajes de ruta de rastreo son mensajes estándar de IBM MQ que contienen un descriptor de mensaje y datos de mensaje. Los datos de mensaje contienen información sobre las actividades realizadas en un mensaje de ruta de rastreo a medida que éste se ha direccionado a través de una red de gestores de colas.

Los mensajes de ruta de rastreo contienen la siguiente información:

Un descriptor de mensaje

Una estructura MQMD, con el campo *Formato* establecido en MQFMT_ADMIN o en MQFMT_EMBEDDED_PCF.

Datos de mensaje

Constan de:

- Un cabecera PCF (MQCFH) y datos del mensaje de ruta de rastreo, si se ha establecido *Formato* en MQFMT, o
- Una cabecera PCF incrustada (MQEPH), los datos del mensaje de ruta de rastreo, y datos de mensaje adicionales especificados por el usuario, si se ha establecido *Formato* en MQFMT_EMBEDDED_PCF.

Cuando se utiliza la aplicación de visualización de ruta de IBM MQ para generar un mensaje de ruta de rastreo, *Formato* se establece en MQFMT_ADMIN.

El contenido de los datos del mensaje de ruta de rastreo viene determinado por el parámetro *Accumulate*, del grupo PCF *TraceRoute*, tal como se indica a continuación:

- Si *Accumulate* se establece en MQRROUTE_ACCUMULATE_NONE, los datos del mensaje de ruta de rastreo contienen el grupo PCF *TraceRoute*.
- Si *Accumulate* se establece en MQRROUTE_ACCUMULATE_IN_MSG o en MQRROUTE_ACCUMULATE_AND_REPLY, los datos del mensaje de ruta de rastreo contienen el grupo PCF *TraceRoute*, y cero o más grupos PCF *Activity*.

La [Tabla 20](#) en la [página 128](#) muestra la estructura de un mensaje de ruta de rastreo.

<i>Tabla 20. Formato de mensaje de ruta de rastreo</i>		
estructura MQMD	Estructura MQEPH de cabecera PCF integrada	Datos de mensaje de ruta de rastreo
Identificador de la estructura	Identificador de la estructura	TraceRoute
Versión de la estructura	Versión de la estructura	Detail
Opciones de informe	Longitud de estructura	Actividades registradas
Tipo de mensaje	Encoding	Actividades no registradas
Hora de caducidad	ID de juego de caracteres	Recuento de discontinuidad
Comentarios	codificado	Núm. máximo de actividades
Encoding	Formato de mensaje	Accumulate
ID de juego de caracteres	Flags	Deliver
codificado	Cabecera PCF (MQCFH)	
Formato de mensaje	Tipo de estructura	
Prioridad	Longitud de estructura	
Persistence	Versión de la estructura	
Identificador de mensaje	Identificador de mandato	
Identificador de correlación	Número de secuencia de mensajes	
Recuento de restituciones	Opciones de control	
Cola de respuestas	Código de terminación	
Gestor de colas de respuestas	Código de razón	
Identificador de usuario	Recuento de parámetros	
Señal de contabilidad		
Datos de identidad de la aplicación		
Tipo de aplicación		
Nombre de la aplicación		
Fecha de transferencia		
Hora de transferencia		
Datos de origen de la aplicación		
Identificador de grupo		
Número de secuencia de mensajes		
Desplazamiento		
Distintivos de mensajes		
Longitud original		

MQMD de mensaje de ruta de rastreo (descriptor de mensaje)

Utilice esta página para ver los valores contenidos en la estructura MQMD de un mensaje de ruta de rastreo.

StrucId

Descripción:	Identificador de estructura.
Tipo de datos:	MQCHAR4.
Valor:	MQMD_STRUC_ID.

Version

Descripción: Número de versión de la estructura.
Tipo de datos: MQLONG.
Valores: **MQMD_VERSION_1.**

Report

Descripción: Opciones para los mensajes de informe.
Tipo de datos: MQLONG.
Valor: Establézcase según los requisitos. Las opciones de informe comunes son las siguientes:

MQRO_DISCARD_MSG

El mensaje se descarta cuando llega a una cola local.

MQRO_PASS_DISCARD_AND_EXPIRY

Cada respuesta (informes de actividad o mensaje de respuesta de ruta de rastreo) tendrá establecida la opción de informe MQRO_DISCARD_MSG y se le pasa la caducidad restante. Esto garantiza que las respuestas no permanezcan en la red de gestores de colas de forma indefinida.

MsgType

Descripción: Tipo de mensaje.
Tipo de datos: MQLONG.
Valor: Si el parámetro *Accumulate* del grupo TraceRoute se especifica como MQROUTE_ACCUMULATE_AND_REPLY, el tipo de mensaje será MQMT_REQUEST.
De lo contrario:
MQMT_DATAGRAM.

Expiry

Descripción: Duración del mensaje.
Tipo de datos: MQLONG.
Valor: Establézcase según los requisitos. Este parámetro puede utilizarse para garantizar que los mensajes de ruta de rastreo no permanezcan en una red de gestores de colas de forma indefinida.

Feedback

Descripción: Código de retorno o de razón.
Tipo de datos: MQLONG.
Valor: **MQFB_NONE.**

Encoding

Descripción: Codificación numérica de datos de mensaje.
Tipo de datos: MQLONG.
Valor: Establézcase según convenga.

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensajes.
Tipo de datos: MQLONG.
Valor: Establézcase según convenga.

Format

Descripción: Nombre de formato de datos de mensaje
Tipo de datos: MQCHAR8.
Valor: **MQFMT_ADMIN**
Mensaje de administración. Ningún dato de usuario sigue el grupo PCF *TraceRoute*.
MQFMT_EMBEDDED_PCF
Mensajes PCF integrados. Los datos de usuario sigue el grupo PCF *TraceRoute*.

Priority

Descripción: Prioridad del mensaje.
Tipo de datos: MQLONG.
Valor: Establézcase según los requisitos.

Persistence

Descripción: Persistencia de mensajes.
Tipo de datos: MQLONG.
Valor: Establézcase según los requisitos.

MsgId

Descripción: Identificador del mensaje.
Tipo de datos: MQBYTE24.
Valor: Establézcase según los requisitos.

CorrelId

Descripción: Identificador de correlación.
Tipo de datos: MQBYTE24.
Valor: Establézcase según los requisitos.

BackoutCount

Descripción: Umbral de restitución.
Tipo de datos: MQLONG.
Valor: 1.

ReplyToQ

Descripción: Nombre de la cola de respuesta.
Tipo de datos: MQCHAR48.

Valores: Establézcase según los requisitos.
Si se establece *MsgType* en MQMT_REQUEST, o si *Report* tiene establecida alguna opción de generación de informes, no debe dejarse en blanco este parámetro.

ReplyToQMgr

Descripción: Nombre del gestor de colas de respuesta.
Tipo de datos: MQCHAR48.
Valor: Establézcase según los requisitos.

UserIdentifier

Descripción: El identificador de usuario de la aplicación que ha originado el mensaje.
Tipo de datos: MQCHAR12.
Valor: Establézcase como se hace normalmente.

AccountingToken

Descripción: Señal de contabilidad que permite que una aplicación cargo el trabajo realizado como resultado del mensaje.
Tipo de datos: MQBYTE32.
Valor: Establézcase como se hace normalmente.

ApplIdentityData

Descripción: Datos de la aplicación relacionados con la identidad.
Tipo de datos: MQCHAR32.
Valores: Establézcase como se hace normalmente.

PutApplType

Descripción: Tipo de aplicación que ha transferido el mensaje.
Tipo de datos: MQLONG.
Valor: Establézcase como se hace normalmente.

PutApplName

Descripción: Nombre de la aplicación que transfiere el mensaje.
Tipo de datos: MQCHAR28.
Valor: Establézcase como se hace normalmente.

PutDate

Descripción: Fecha en que se transfirió el mensaje.
Tipo de datos: MQCHAR8.
Valor: Establézcase como se hace normalmente.

PutTime

Descripción: Hora en que se transfirió el mensaje.

Tipo de datos: MQCHAR8.
Valor: Establézcase como se hace normalmente.

ApplOriginData

Descripción: Datos de la aplicación relacionados con el origen.
Tipo de datos: MQCHAR4.
Valor: Establézcase como se hace normalmente.

MQEPH de mensaje de ruta de rastreo (cabecera PCF incorporada)

Utilice esta página para ver los valores contenidos en la estructura MQEPH de un mensaje de ruta de rastreo.

La estructura MQEPH contiene una descripción tanto de la información PCF que acompaña a los datos de mensaje de un mensaje de ruta de rastreo, como de los datos de mensaje de aplicación que le siguen. Una estructura MQEPH sólo se utiliza si los datos del mensaje de usuario adicional siguen el grupo PCF TraceRoute.

Para un mensaje de ruta de rastreo, la estructura MQEPH contiene los valores siguientes:

StrucId

Descripción: Identificador de estructura.
Tipo de datos: MQCHAR4.
Valor: MQEPH_STRUC_ID.

Version

Descripción: Número de versión de la estructura.
Tipo de datos: MQLONG.
Valores: MQEPH_VERSION_1.

StrucLength

Descripción: Longitud de la estructura.
Tipo de datos: MQLONG.
Valor: Longitud total de la estructura, incluidas las estructuras de parámetros PCF que le siguen.

Encoding

Descripción: Codificación numérica de los datos de mensaje que siguen a la última estructura de parámetros PCF.
Tipo de datos: MQLONG.
Valor: La codificación de los datos del mensaje.

CodedCharSetId

Descripción: Identificador de juego de caracteres de los datos de mensaje que siguen a la última estructura de parámetros PCF.
Tipo de datos: MQLONG.
Valor: El juego de caracteres de los datos del mensaje.

Format

Descripción:	Nombre del formato de los datos de mensaje que siguen a la última estructura de parámetros PCF.
Tipo de datos:	MQCHAR8.
Valor:	El nombre del formato de los datos del mensaje.

Flags

Descripción:	Distintivos que especifican los atributos de la estructura, o controlan su proceso.
Tipo de datos:	MQLONG.
Valor:	MQEPH_NONE No se especifica ningún distintivo. MQEPH_CCSID_EMBEDDED Especifica que el conjunto de caracteres de los parámetros que contienen los datos de caracteres se especifica individualmente, dentro del campo <i>CodedCharSetId</i> , de cada estructura.

PCFHeader

Descripción:	Cabecera de formato de mandato programable
Tipo de datos:	MQCFH.
Valor:	Consulte “MQCFH de mensaje de ruta de rastreo (cabecera PCF)” en la página 133.

MQCFH de mensaje de ruta de rastreo (cabecera PCF)

Utilice esta página para ver los valores PCF contenidos en la estructura MQCFH de un mensaje de ruta de rastreo.

Para un mensaje de ruta de rastreo, la estructura MQCFH contiene los valores siguientes:

Type

Descripción:	Tipo de estructura que identifica el contenido del mensaje.
Tipo de datos:	MQLONG.
Valor:	MQCFT_TRACE_ROUTE El mensaje es un mensaje de ruta de rastreo.

StrucLength

Descripción:	Longitud de la estructura.
Tipo de datos:	MQLONG.
Valor:	MQCFH_STRUC_LENGTH Longitud en bytes de la estructura MQCFH.

Version

Descripción:	Número de versión de la estructura.
Tipo de datos:	MQLONG.
Valores:	MQCFH_VERSION_3

Command

Descripción:	Identificador de mandato. Identifica la categoría del mensaje.
Tipo de datos:	MQLONG.
Valores:	MQCMD_TRACE_ROUTE Mensaje de ruta de rastreo.

MsgSeqNumber

Descripción:	Número de secuencia de mensaje. Éste es el número de secuencia del mensaje dentro de un grupo de mensajes relacionados.
Tipo de datos:	MQLONG.
Valores:	1.

Control

Descripción:	Opciones de control.
Tipo de datos:	MQLONG.
Valores:	MQCFC_LAST.

CompCode

Descripción:	Código de terminación.
Tipo de datos:	MQLONG.
Valores:	MQCC_OK.

Reason

Descripción:	Código de razón que califica el código de terminación.
Tipo de datos:	MQLONG.
Valores:	MQRC_NONE.

ParameterCount

Descripción:	Recuento de las estructuras de parámetro. Es el número de estructuras de parámetro que siguen la estructura MQCFH. Una estructura de grupo (MQCFGR) y las estructuras de parámetro que incluye, se consideran como una estructura única.
Tipo de datos:	MQLONG.
Valores:	1 o mayor.

Datos de mensaje de ruta de rastreo

Utilice esta página para ver los parámetros de que consta la parte del grupo PCF *TraceRoute* de los datos del mensaje de ruta de rastreo.

El contenido de los datos del mensaje de ruta de rastreo depende del parámetro *Accumulate* del grupo PCF *TraceRoute*. Los datos del mensaje de ruta de rastreo constan del grupo PCF *TraceRoute*, y cero o más grupos PCF *Activity*. En este tema se describe el grupo PCF *TraceRoute* de forma detallada. Consulte la información relacionada para obtener detalles del grupo PCF *Activity*.

Los datos del mensaje de ruta de rastreo contienen los parámetros siguientes:

TraceRoute

Descripción:	Parámetros agrupados que especifican los atributos del mensaje de ruta de rastreo. Para un mensaje de ruta de rastreo, algunos de estos parámetros se pueden modificar para controlar la forma en que se procesa.
Identificador:	MQGACF_TRACE_ROUTE.
Tipo de datos:	MQCFGR.
Contenidos en el grupo PCF:	Ninguna.
Parámetros del grupo:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>

Detail

Descripción:	El nivel de detalle que se registrará para la actividad.
Identificador:	MQIACF_ROUTE_DETAIL.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .
Valores:	MQROUTE_DETAIL_LOW Se registran las actividades realizadas por una aplicación definida por el usuario. MQROUTE_DETAIL_MEDIUM Se registran las actividades especificadas en MQROUTE_DETAIL_LOW. Además, se registran las actividades realizadas por los MCA. MQROUTE_DETAIL_HIGH Se registran las actividades especificadas en MQROUTE_DETAIL_LOW y MQROUTE_DETAIL_MEDIUM. Los MCA no registran ninguna otra información de actividad en este nivel de detalle. Esta opción sólo está disponible para las aplicaciones escritas por el usuario que vayan a registrar información de actividad adicional.

RecordedActivities

Descripción:	El número de actividades que el mensaje de ruta de rastreo ha provocado, donde se ha registrado información.
Identificador:	MQIACF_RECORDED_ACTIVITIES.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .

UnrecordedActivities

Descripción:	El número de actividades que el mensaje de ruta de rastreo ha provocado, donde no se ha registrado ninguna información.
Identificador:	MQIACF_UNRECORDED_ACTIVITIES.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .

DiscontinuityCount

Descripción:	El número de veces que se ha recibido un mensaje de ruta de rastreo procedente de un gestor de colas que no dé soporte a la mensajería de ruta de rastreo.
Identificador:	MQIACF_DISCONTINUITY_COUNT.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .

MaxActivities

Descripción:	El número máximo de actividades en que puede estar implicado el mensaje de ruta de rastreo antes de que deje de procesarse.
Identificador:	MQIACF_MAX_ACTIVITIES.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .
Valor:	Un entero positivo El número máximo de actividades. MQROUTE_UNLIMITED_ACTIVITIES Un número ilimitado de actividades.

Accumulate

Descripción:	Especifica si la información de actividad se acumula en el mensaje de ruta de rastreo, y si se genera un mensaje de respuesta que contenga la información de actividad acumulada antes de que se descarte el mensaje de ruta de rastreo, o de que se coloque en una cola de transmisión distinta.
Identificador:	MQIACF_ROUTE_ACCUMULATION.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .

Valor:	MQRROUTE_ACCUMULATE_NONE La información de actividad no se acumula en los datos de mensaje del mensaje de ruta de rastreo.
	MQRROUTE_ACCUMULATE_IN_MSG La información de actividad se acumula en los datos de mensaje del mensaje de ruta de rastreo.
	MQRROUTE_ACCUMULATE_AND_REPLY La información de actividad se acumula en los datos de mensaje del mensaje de ruta de rastreo, y se generará un mensaje de respuesta de ruta de rastreo.

Forward

Descripción:	Especifica los gestores de colas a los que se puede reenviar el mensaje de ruta de rastreo. Al determinar si reenviar un mensaje a un gestor de colas remoto, los gestores de colas utilizan el algoritmo que se describe en Forwarding .
Identificador:	MQIACF_ROUTE_FORWARDING.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .
Valor:	MQRROUTE_FORWARD_IF_SUPPORTED El mensaje de ruta de rastreo sólo se reenvía a los gestores de colas que acepten el valor del parámetro <i>Deliver</i> del grupo <i>TraceRoute</i> .
	MQRROUTE_FORWARD_ALL El mensaje de ruta de rastreo se reenvía a cualquier gestor de colas, independientemente de si se acepta el valor del parámetro <i>Deliver</i> .

Deliver

Descripción:	Especifica la acción que se debe efectuar si el mensaje de ruta de rastreo llega a su cola de destino de forma satisfactoria.
Identificador:	MQIACF_ROUTE_DELIVERY.
Tipo de datos:	MQCFIN.
Contenidos en el grupo PCF:	<i>TraceRoute</i> .
Valor:	MQRROUTE_DELIVER_YES Al llegar, el mensaje de ruta de rastreo se coloca en la cola de destino. Cualquier aplicación que realice una operación de tipo get destructiva en la cola de destino puede recibir el mensaje de ruta de rastreo.
	MQRROUTE_DELIVER_NO El mensaje de ruta de rastreo se descarta cuando llega.

Referencia de mensaje de respuesta de ruta de rastreo

Utilice esta página para obtener una visión general del formato de mensaje de respuesta de ruta de rastreo. Los datos de mensaje de respuesta de ruta de rastreo son un duplicado de los datos de mensaje de ruta de rastreo del mensaje de ruta de rastreo para el que se han generado.

Formato de mensaje de respuesta de ruta de rastreo

Los mensajes de respuesta de ruta de rastreo son mensajes estándar de IBM MQ que contienen un descriptor de mensaje y datos de mensaje. Los datos de mensaje contienen información sobre las

actividades realizadas en un mensaje de ruta de rastreo a medida que éste se ha direccionado a través de una red de gestores de colas.

Los mensajes de respuesta de ruta de rastreo contienen la información siguiente:

Un descriptor de mensaje

Una estructura MQMD

Datos de mensaje

Un cabecera PCF (MQCFH) y datos de mensaje de respuesta de ruta de rastreo

Los datos de mensaje de respuesta de ruta de rastreo consta de uno o más grupos PCF *Activity*.

Cuando un mensaje de ruta de rastreo llega a su cola de destino, se puede generar un mensaje de respuesta de ruta de rastreo que contiene una copia de la información de actividad del mensaje de ruta de rastreo. El mensaje de respuesta de ruta de rastreo se entregará en una cola de respuesta o una cola del sistema.

En la Tabla 21 en la página 138 se muestra la estructura de un mensaje de respuesta de ruta de rastreo, incluidos los parámetros que sólo se devuelven bajo ciertas condiciones.

Tabla 21. Formato de mensaje de respuesta de ruta de rastreo

estructura MQMD	Estructura MQCFH de cabecera PCF	Datos del mensaje de respuesta de ruta de rastreo
Identificador de la estructura	Cabecera PCF (MQCFH)	Actividad
Versión de la estructura	Tipo de estructura	Nombre de la aplicación de la actividad
Opciones de informe	Longitud de estructura	Nombre de la aplicación de la actividad
Tipo de mensaje	Versión de la estructura	Tipo de aplicación de la actividad
Hora de caducidad	Identificador de mandato	Descripción de la actividad
Comentarios	Número de secuencia de mensajes	Operación
Encoding	Opciones de control	Tipo de operación
ID de juego de caracteres codificado	Código de terminación	Fecha de operación
Formato de mensaje	Código de razón	Hora de operación
Prioridad	Recuento de parámetros	Mensaje
Persistence		Longitud del mensaje
Identificador de mensaje		MQMD
Identificador de correlación		EmbeddedMQMD
Recuento de restituciones		Nombre del gestor de colas
Cola de respuestas		Nombre del grupo de compartición de colas
Gestor de colas de respuestas		Nombre de cola ^{1 2 3}
Identificador de usuario		Nombre de cola resuelto ^{1 3}
Señal de contabilidad		Nombre de cola remota ³
Datos de identidad de la aplicación		Gestor de colas remoto-nombre ^{2 3 4 5}
Tipo de aplicación		Realimentación ²
Nombre de la aplicación		Nombre de canal ^{4 5}
Fecha de transferencia		Tipo de canal ^{4 5}
Hora de transferencia		Nombre de cola de transmisión ⁵
Datos de origen de la aplicación		TraceRoute
Identificador de grupo		Detail
Número de secuencia de mensajes		Actividades registradas
Desplazamiento		Actividades no registradas
Distintivos de mensajes		Recuento de discontinuidad
Longitud original		Núm. máximo de actividades
		Accumulate
		Deliver

Tabla 21. Formato de mensaje de respuesta de ruta de rastreo (continuación)

estructura MQMD	Estructura MQCFH de cabecera PCF	Datos del mensaje de respuesta de ruta de rastreo
<p>Nota:</p> <ol style="list-style-type: none"> 1. Se devuelve para las operaciones get y browse. 2. Se devuelve para las operaciones Discard. 3. Se devuelve para las operaciones Put, Put Reply y Put Report. 4. Se devuelve para las operaciones Receive. 5. Se devuelve para las operaciones Send. 		

MQMD de mensaje de respuesta de ruta de rastreo (descriptor de mensaje)

Utilice esta página para ver los valores contenidos en la estructura MQMD de un mensaje de respuesta de ruta de rastreo.

Para un mensaje de respuesta de ruta de rastreo, la estructura MQMD contiene los parámetros descritos en Descriptor de mensaje de informe de actividad. Algunos de los valores de parámetro de un descriptor de mensaje de respuesta de ruta de rastreo son diferentes de los de un descriptor de mensaje de informe de actividad, tal como se indica a continuación:

MsgType

Descripción: Tipo de mensaje.
 Tipo de datos: MQLONG.
 Valor: **MQMT_REPLY**

Feedback

Descripción: Código de retorno o de razón.
 Tipo de datos: MQLONG.
 Valor: **MQFB_NONE**

Encoding

Descripción: Codificación numérica de datos de mensaje.
 Tipo de datos: MQLONG.
 Valor: Se copia de un descriptor de mensaje de ruta de rastreo.

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensajes.
 Tipo de datos: MQLONG.
 Valor: Se copia de un descriptor de mensaje de ruta de rastreo.

Format

Descripción: Nombre de formato de datos de mensaje
 Tipo de datos: MQCHAR8.
 Valor: **MQFMT_ADMIN**
 Mensaje de administración.

MQCFH de mensaje de respuesta de ruta de rastreo (cabecera PCF)

Utilice esta página para ver los valores PCF contenidos en la estructura MQCFH de un mensaje de respuesta de ruta de rastreo.

La cabecera PCF (MQCFH) para un mensaje de respuesta de ruta de rastreo es la misma que para un mensaje de ruta de rastreo.

Datos del mensaje de respuesta de ruta de rastreo

Los datos de mensaje de respuesta de ruta de rastreo son un duplicado de los datos de mensaje de ruta de rastreo del mensaje de ruta de rastreo para el que se han generado.

Los datos de mensaje de respuesta de ruta de rastreo contienen uno o más grupos PCF *Activity*. Los parámetros se describen en [“Datos de mensaje de informe de actividad”](#) en la página 110.

Mensajes de contabilidad y estadísticas


Los gestores de colas generan mensajes de contabilidad y estadísticas para registrar información sobre las operaciones MQI realizadas por las aplicaciones IBM MQd, o para registrar información sobre las actividades que se producen en un sistema IBM MQ.

Mensajes de contabilidad

Los mensajes de contabilidad se utilizan para registrar información sobre las operaciones de MQI realizadas por las aplicaciones de IBM MQ; consulte [“Mensajes de contabilidad”](#) en la página 140.

Mensajes de estadísticas

Los mensajes de estadísticas se utilizan para registrar información sobre las actividades que se producen en un sistema IBM MQ; consulte [“Mensajes de estadísticas”](#) en la página 144. Algunas actividades registradas en los mensajes de estadísticas están relacionadas con operaciones internas del gestor de colas.

 Los mensajes de contabilidad y de estadísticas tal como se describen aquí no están disponibles en IBM MQ for z/OS, pero existe funcionalidad equivalente disponible a través del recurso de gestión del sistema (SMF).

Los mensajes de contabilidad y de estadísticas se entregan a una de las dos colas del sistema. Las aplicaciones de usuario pueden recuperar los mensajes de estas colas del sistema y utilizar la información registrada para varias finalidades:

- Realizar la contabilidad del uso de recursos de la aplicación.
- Registrar la actividad de la aplicación.
- Planificar la capacidad.
- Detectar problemas en la red del gestor de colas.
- Ayudar en la determinación de las causas de problemas de la red del gestor de colas.
- Mejorar la eficacia de la red del gestor de colas.
- Familiarizarse con la ejecución de la red del gestor de colas.
- Comprobar que la red del gestor de colas funciona correctamente.

Conceptos relacionados

[“Using System Management Facility”](#) en la página 328

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

Mensajes de contabilidad

Los mensajes de contabilidad registran información sobre las operaciones MQI realizadas por las aplicaciones IBM MQ. Un mensaje de contabilidad es un mensaje PCF que contiene varias estructuras PCF.

Cuando una aplicación se desconecta de un gestor de colas, se genera un mensaje de contabilidad, que se entrega en la cola de contabilidad del sistema (SYSTEM.ADMIN.ACCOUNTING.QUEUE). Para las

aplicaciones IBM MQ de larga ejecución, se generan mensajes de contabilidad intermedios tal como se indica a continuación:

- Cuando el tiempo transcurrido desde que se estableció la conexión excede el intervalo configurado.
- Cuando el tiempo transcurrido desde el último mensaje de contabilidad intermedio excede el intervalo configurado.

Los mensajes de contabilidad se encuentran en las categorías siguientes:

Mensajes de contabilidad MQI

Los mensajes de contabilidad MQI contienen información relacionada con el número de llamadas MQI realizadas mediante una conexión a un gestor de colas.

Mensajes de contabilidad de cola

Los mensajes de contabilidad MQI contienen información relacionada con el número de llamadas MQI realizadas mediante las conexiones a un gestor de colas, agrupadas por cola.

Cada mensaje de contabilidad de cola puede contener hasta 100 registros, en que cada registro es relativo a una actividad que la aplicación realiza con respecto a una cola específica.

Los mensajes de contabilidad se registran sólo para las colas locales. Si una aplicación efectúa una llamada MQI en una cola alias, los datos de contabilidad se registran en la cola base y, para una cola remota, los datos de contabilidad se registran en la cola de transmisión.

Nota: Puesto que la información está relacionada con las operaciones MQI realizadas por las aplicaciones IBM MQ, la información no incluye las operaciones relacionadas con una cola de modalidad continua, a menos que la operación esté directamente en esa cola por parte de la aplicación.

Referencia relacionada

[“datos de mensaje de contabilidad de MQI” en la página 160](#)

Utilice esta página para ver la estructura de un mensaje de contabilidad de MQI.

[“Datos de mensajes de contabilidad de cola” en la página 171](#)

Utilice esta página para ver la estructura de un mensaje de contabilidad de cola.

Formato de mensaje de contabilidad

Los mensajes de contabilidad constan de un conjunto de campos PCF que contienen un descriptor de mensaje y los datos de mensaje.

Descriptor de mensaje

- Un MQMD (descriptor de mensaje) de mensaje de contabilidad

Datos de mensaje de contabilidad

- Una MQCFH (cabecera PCF) del mensaje de contabilidad
- Los datos del mensaje de contabilidad que se devuelven siempre
- Los datos del mensaje de contabilidad que se devuelven si están disponibles

La MQCFH (cabecera PCF) de mensaje de contabilidad contiene información sobre la aplicación, y el intervalo para el que se han registrado los datos contables.

Los datos de mensaje de contabilidad constan de parámetros PCF que almacenan la información sobre la contabilidad. El contenido de los mensajes de contabilidad depende de la categoría de mensaje, tal como se indica a continuación:

Mensaje de contabilidad MQI

Los datos de mensaje de contabilidad MQI constan de varios parámetros PCF, pero de ningún grupo PCF.

Mensaje de contabilidad de cola

Los datos del mensaje de contabilidad de cola constan de varios parámetros PCF, y en el rango de 1 a 100 *QAccountingData* grupos PCF.

Hay un grupo PCF *QAccountingData* para cada cola que tenía datos contables recopilados. Si una aplicación accede a más de 100 colas, se generan varios mensajes de contabilidad. Cada mensaje tiene el *SeqNumber* en el MQCFH (cabecera PCF) actualizado en consecuencia, y el último mensaje en la secuencia tiene el parámetro *Control* en el MQCFH especificado como MQCFC_LAST.

Recopilación de información de contabilidad

Utilice los atributos de la cola y del gestor de colas para controlar la recopilación de información de contabilidad. También puede utilizar las opciones MQCONNX para controlar la recopilación en el nivel de conexión.

Control de la recopilación de información de contabilidad de MQI

Utilice el atributo de gestor de colas ACCTMQI para controlar la recopilación de la información de contabilidad de MQI.

Para cambiar el valor de este atributo, utilice el mandato MQSC ALTER QMGR, y especifique el parámetro ACCTMQI. Los mensajes de contabilidad sólo se generan para las conexiones que empiecen después de habilitarse la contabilidad. El parámetro **ACCTMQI** puede tener los valores siguientes:

ON

La información de contabilidad MQI se recopila para cada conexión que se efectúa con el gestor de colas.

OFF

No se recopila ninguna información de contabilidad MQI. Éste es el valor predeterminado.

Por ejemplo, para habilitar la recopilación de información de contabilidad MQI, utilice el mandato MQSC siguiente:

```
ALTER QMGR ACCTMQI(ON)
```

Información de contabilidad de cola

Utilice el atributo de cola ACCTQ y el atributo de gestor de colas ACCTQ para controlar la recopilación de información de contabilidad de la cola.

Para cambiar el valor del atributo de cola, utilice el mandato MQSC, ALTER QLOCAL, y especifique el parámetro ACCTQ. Los mensajes de contabilidad sólo se generan para las conexiones que empiecen después de habilitarse la contabilidad. Tenga en cuenta que los cambios en este valor solo surten efecto para las conexiones con el gestor de colas que se producen después del cambio en el atributo.

El atributo de cola ACCTQ puede tener los valores siguientes:

ON

La información de contabilidad de esta cola se recopila para cada conexión que exista con el gestor de colas que abre la cola.

OFF

Para esta cola no se recopila la información de contabilidad de cola.

QMGR

La recopilación de información de contabilidad de cola para esta cola se controla en función del valor del atributo de gestor de colas ACCTQ. Éste es el valor predeterminado.

Para cambiar el valor del atributo de gestor de colas, utilice el mandato MQSC ALTER QMGR, y especifique el parámetro ACCTQ. El atributo de gestor de colas ACCTQ puede tener los valores siguientes:

ON

La información de contabilidad de cola se recopila para las colas que tengan el atributo de cola ACCTQ establecido como QMGR.

OFF

La información de contabilidad de cola no se recopila para las colas que tengan el atributo de cola ACCTQ establecido como QMGR. Éste es el valor predeterminado.

NINGUNO

La recopilación de información de contabilidad de cola está inhabilitada para todas las colas, independientemente del atributo de cola ACCTQ.

Si el atributo de gestor de colas, ACCTQ, está establecido en NONE, la recopilación de información de contabilidad de cola está inhabilitada para todas las colas, independientemente del atributo de cola ACCTQ.

Por ejemplo, para habilitar la recopilación de información de contabilidad para la cola, Q1, utilice el mandato MQSC siguiente:

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

Para habilitar la recopilación de información de contabilidad para todas las colas que especifiquen el atributo de cola ACCTQ como QMGR, utilice el mandato MQSC siguiente:

```
ALTER QMGR ACCTQ(ON)
```

Opciones MQCONNX para controlar la recopilación de información de contabilidad

Utilice el parámetro **ConnectOpts** en la llamada MQCONNX para modificar la recopilación tanto de la información MQI como de la información de contabilidad de la cola en el nivel de conexión, alterando temporalmente los valores efectivos de los atributos de gestor de colas ACCTMQI y ACCTQ.

El parámetro **ConnectOpts** puede tener los valores siguientes:

MQCNO_ACCOUNTING_MQI_ENABLED

Si el valor del atributo de gestor de colas ACCTMQI se especifica como OFF, se habilita la contabilidad MQI para esta conexión. Esto equivale al especificar el atributo de gestor de colas ACCTMQI como ON.

Si el valor del atributo de gestor de colas ACCTMQI no se especifica como OFF, este atributo no tiene efecto alguno.

MQCNO_ACCOUNTING_MQI_DISABLED

Si el valor del atributo de gestor de colas ACCTMQI se especifica como ON, se inhabilita la contabilidad MQI para esta conexión. Esto equivale al especificar el atributo de gestor de colas ACCTMQI como OFF.

Si el valor del atributo de gestor de colas ACCTMQI no se especifica como ON, este atributo no tiene efecto alguno.

MQCNO_ACCOUNTING_Q_ENABLED

Si el valor del atributo de gestor de colas ACCTQ se especifica como OFF, se habilita la contabilidad de la cola para esta conexión. Todas las colas en las que se haya especificado ACCTQ como QMGR, están habilitadas para la contabilidad de la cola. Esto equivale al especificar el atributo de gestor de colas ACCTQ como ON.

Si el valor del atributo de gestor de colas ACCTQ no se especifica como OFF, este atributo no tiene efecto alguno.

MQCNO_ACCOUNTING_Q_DISABLED

Si el valor del atributo de gestor de colas ACCTQ se especifica como ON, se inhabilita la contabilidad de la cola para esta conexión. Esto equivale al especificar el atributo de gestor de colas ACCTQ como OFF.

Si el valor del atributo de gestor de colas ACCTQ no se especifica como ON, este atributo no tiene efecto alguno.

De forma predeterminada, estas alteraciones temporales están inhabilitadas. Para habilitarlas, establezca el atributo de gestor de colas ACCTCONO en ENABLED. Para habilitar las alteraciones temporales de contabilidad para las conexiones individuales, utilice el mandato MQSC siguiente:

```
ALTER QMGR ACCTCONO(ENABLED)
```

Generación de mensajes de contabilidad cuando las aplicaciones se desconectan

Los mensajes de contabilidad se generan cuando una aplicación se desconecta del gestor de colas. También se graban mensajes de contabilidad intermedios para las aplicaciones IBM MQ de larga ejecución.

Los mensajes de contabilidad se generan de una de las maneras siguientes, cuando se desconecta una aplicación:

- La aplicación emite una llamada MQDISC.
- El gestor de colas reconoce que la aplicación ha finalizado.

Se graban mensajes de contabilidad intermedios para las aplicaciones IBM MQ de larga ejecución cuando el intervalo desde que se estableció la conexión, o desde que se ha grabado el último mensaje de contabilidad intermedio, excede el intervalo configurado. El atributo de gestor de colas ACCTINT especifica el tiempo, en segundos, tras el cual se pueden grabar automáticamente mensajes de contabilidad intermedios. Los mensajes de contabilidad se generan sólo cuando la aplicación interactúa con el gestor de colas, de modo que las aplicaciones que permanezcan conectadas al gestor de colas durante largos períodos sin ejecutar las solicitudes MQI, no generarán mensajes de contabilidad hasta que se ejecute la primera solicitud MQI posterior a la finalización del intervalo de contabilidad.

El valor de contabilidad predeterminado es 1800 segundos (30 minutos). Por ejemplo, para cambiar el intervalo de contabilidad a 900 segundos (15 minutos), utilice el mandato MQSC siguiente:

```
ALTER QMGR ACCTINT(900)
```

Mensajes de estadísticas

Los mensajes de estadísticas registran información sobre las actividades que se llevan a cabo en un sistema IBM MQ. Un mensaje de estadísticas es un mensaje PCF que contiene varias estructuras PCF.

Los mensajes de estadísticas se entregan a la cola de sistema (SYSTEM.ADMIN.STATISTICS.QUEUE) a intervalos configurados, siempre que hay actividad.

Los mensajes de estadísticas se encuentran en las categorías siguientes:

Mensajes de estadísticas MQI

Los mensajes de estadísticas MQI contienen información relacionada con el número de llamadas MQI realizadas durante un intervalo configurado. Por ejemplo, la información puede incluir el número de llamadas MQI que ha emitido un gestor de colas.

Mensajes de estadísticas de cola

Los mensajes de estadísticas de cola contienen información relacionada con la actividad de una cola durante un intervalo configurado. La información incluye el número de mensajes colocados en, y recuperados de, la cola, y el número total de bytes procesados por una cola.

Cada mensaje de estadísticas de cola puede contener hasta 100 registros, en que cada registro es relativo a la actividad según la cola para la que se hayan recopilado las estadísticas.

Los mensajes de estadísticas se registran sólo para las colas locales. Si una aplicación efectúa una llamada MQI en una cola alias, los datos de estadísticas se registran en la cola base y, para una cola remota, los datos de estadísticas se registran en la cola de transmisión.

Mensajes de estadísticas de canal

Los mensajes de estadísticas de canal contienen información relacionada con la actividad de un canal durante un intervalo configurado. Por ejemplo la información puede ser el número de mensajes que ha transferido el canal, o el número de bytes que ha transferido el canal.

Cada mensaje de estadísticas de canal puede contener hasta 100 registros, en que cada registro es relativo a la actividad según el canal para el que se hayan recopilado las estadísticas.

Nota: A medida que la información se relaciona con actividades que se producen en un sistema IBM MQ , la información incluye operaciones relacionadas con colas de modalidad continua.

Referencia relacionada

“Control de la recopilación de estadísticas de MQI” en la página 146

Utilice el atributo del gestor de colas STATMQI para controlar la recopilación de la información de estadística de MQI.

“Control de la recopilación de estadísticas de cola” en la página 146

Utilice el atributo de cola STATQ y el atributo de gestor de colas STATQ para controlar la recopilación de información de estadísticas de cola.

“Control de la recopilación de estadísticas de canal” en la página 147

Utilice el atributo de canal STATCHL para controlar la recopilación de información de estadísticas de canal. También puede establecer atributos de gestor de colas para controlar la recopilación de información.

Formato de los mensajes de estadísticas

Los mensajes de estadísticas constan de un conjunto de campos PCF que contienen un descriptor de mensaje y datos de mensaje.

Descriptor de mensaje

- Un MQMD (descriptor de mensaje) de mensaje de estadísticas

Datos de mensaje de contabilidad

- Una MQCFH (cabecera PCF) de mensaje de estadísticas
- Los datos del mensaje de estadísticas que se devuelven siempre
- Los datos del mensaje de estadísticas que se devuelven si están disponibles

La MQCFH (cabecera PCF) de mensaje de estadísticas contiene información sobre el intervalo para el que se han registrado los datos de estadísticas.

Los datos de mensaje de estadísticas constan de parámetros PCF que almacenan la información sobre las estadísticas. El contenido de los mensajes de estadísticas depende de la categoría de mensaje, tal como se indica a continuación:

Mensaje de estadísticas MQI

Los datos de mensaje de estadísticas MQI constan de varios parámetros PCF, pero de ningún grupo PCF.

Mensaje de estadísticas de cola

Los datos de mensaje de estadísticas de cola constan de varios parámetros PCF, y de los grupos PCF *QStatisticsData* que están en el rango de 1 a 100.

Hay un grupo PCF *QStatisticsData* para cada cola que haya estado activa en el intervalo. Si había más de 100 colas activas en el intervalo, se habrán generado varios mensajes de estadísticas. Cada mensaje tiene el *SeqNumber* en el MQCFH (cabecera PCF) actualizado en consecuencia, y el último mensaje en la secuencia tiene el parámetro *Control* en el MQCFH especificado como MQCFC_LAST.

Mensaje de estadísticas de canal

Los datos de mensaje de canal constan de varios parámetros PCF, y de los grupos PCF *ChlStatisticsData* que están en el rango de 1 a 100.

Hay un grupo PCF *ChlStatisticsData* para cada canal que haya estado activo en el intervalo. Si había más de 100 canales activos en el intervalo, se habrán generado varios mensajes de estadísticas. Cada mensaje tiene el *SeqNumber* en el MQCFH (cabecera PCF) actualizado en consecuencia, y el último mensaje en la secuencia tiene el parámetro *Control* en el MQCFH especificado como MQCFC_LAST.

Recopilación de información de estadísticas

Utilice atributos de cola, de gestor de colas y de canal para controlar la recopilación de información de estadísticas.

Control de la recopilación de estadísticas de MQI

Utilice el atributo del gestor de colas **STATMQI** para controlar la recopilación de la información de estadística de MQI.

Para cambiar el valor de este atributo, utilice el mandato **MQSC**, **ALTER QMGR** y especifique el parámetro **STATMQI**. Los mensajes de estadísticas se generan solamente para las colas que se abran después de que se haya habilitado la recopilación de estadísticas. El parámetro **STATMQI** puede tener los valores siguientes:

ON

La información de estadísticas MQI se recopila para cada conexión que se efectúa con el gestor de colas.

OFF

No se recopila ninguna información de estadísticas MQI. Éste es el valor predeterminado.

Por ejemplo, para habilitar la recopilación de información de estadísticas MQI, utilice el mandato **MQSC** siguiente:

```
ALTER QMGR STATMQI(ON)
```

Control de la recopilación de estadísticas de cola

Utilice el atributo de cola **STATQ** y el atributo de gestor de colas **STATQ** para controlar la recopilación de información de estadísticas de cola.

Puede habilitar o inhabilitar la recopilación de información de estadísticas de cola para colas individuales o para varias colas. Para controlar colas individuales, establezca el atributo de cola **STATQ**. Puede habilitar o inhabilitar la recopilación de información de estadísticas de cola en el nivel de gestor de colas, mediante el atributo de gestor de colas **STATQ**. Para todas las colas que tengan el atributo de cola **STATQ** especificado con el valor **QMGR**, la recopilación de información de estadísticas de cola se controla en el nivel del gestor de colas.

Las estadísticas de colas se incrementan únicamente para las operaciones de los manejadores de objetos de IBM MQ MQI que se han abierto después de que se haya habilitado la recopilación de estadísticas.

Los mensajes de estadísticas de colas se generan solamente para las colas para las que se han recopilado datos de estadísticas en el período de tiempo anterior.

La misma cola puede tener varias operaciones put y get a través de varios manejadores de objetos. Es posible que algunos manejadores de objetos se hayan abierto antes de habilitar la recopilación de estadísticas, pero otros se han abierto después. Por lo tanto, es posible que las estadísticas de cola registren la actividad de algunas operaciones put y get, pero no de todas.

Para asegurarse de que las estadísticas de cola registran la actividad de todas las aplicaciones, debe cerrar y volver a abrir los manejadores de objetos nuevos de la cola, o colas, que va a supervisar. La mejor forma de lograrlo es finalizar y reiniciar todas las aplicaciones después de habilitar la recopilación de estadísticas.

Para cambiar el valor del atributo de cola **STATQ**, utilice el mandato **MQSC** **ALTER QLOCAL**, y especifique el parámetro **STATQ**. El atributo de cola **STATQ** puede tener los valores siguientes:

ON

La información de estadísticas de cola se recopila para cada conexión que exista con el gestor de colas que abre la cola.

OFF

Para esta cola no se recopila la información de estadísticas de cola.

QMGR

La recopilación de información de estadísticas de cola para esta cola se controla en función del valor del atributo de gestor de colas, **STATQ**. Éste es el valor predeterminado.

Para cambiar el valor del atributo de gestor de colas **STATQ**, utilice el mandato **MQSC** **ALTER QMGR**, y especifique el parámetro **STATQ**. El atributo de gestor de colas **STATQ** puede tener los valores siguientes:

ON

La información de estadísticas de cola se recopila para las colas que tengan el atributo de cola STATQ establecido como QMGR.

OFF

La información de estadísticas de cola no se recopila para las colas que tengan el atributo de cola STATQ establecido como QMGR. Éste es el valor predeterminado.

NINGUNO

La recopilación de información de estadísticas de cola está inhabilitada para todas las colas, independientemente del atributo de cola STATQ.

Si el atributo de gestor de colas STATQ está establecido en NONE, la recopilación de información de estadísticas de cola está inhabilitada para todas las colas, independientemente del atributo de cola STATQ.

Por ejemplo, para habilitar la recopilación de información de estadísticas para la cola, Q1, utilice el mandato MQSC siguiente:

```
ALTER QLOCAL(Q1) STATQ(ON)
```

Para habilitar la recopilación de información de estadísticas para todas las colas que especifiquen el atributo de cola STATQ como QMGR, utilice el mandato MQSC siguiente:

```
ALTER QMGR STATQ(ON)
```

Control de la recopilación de estadísticas de canal

Utilice el atributo de canal STATCHL para controlar la recopilación de información de estadísticas de canal. También puede establecer atributos de gestor de colas para controlar la recopilación de información.

Puede habilitar o inhabilitar la recopilación de información de estadísticas de canal para canales individuales o para varios canales. Para controlar canales individuales, debe establecer el atributo de canal STATCHL, para habilitar o inhabilitar la recopilación de información de estadísticas de canal. Para controlar varios canales juntos, debe habilitar o inhabilitar la recopilación de información de estadísticas de canal de a nivel de gestor de colas, mediante el atributo de gestor de colas STATCHL. Para todos los canales que tengan el atributo de canal STATCHL especificado con el valor QMGR, la recopilación de información de estadísticas de canal se controla en el nivel del gestor de colas.

Los canales de clúster emisor definidos automáticamente no son objetos de IBM MQ, por lo que no tienen atributos como los objetos de canal. Para controlar los canales de clúster emisor definidos automáticamente, utilice el atributo de gestor de colas STATACLS. Este atributo determina si los canales de clúster emisor que se definen automáticamente en un gestor de colas están habilitados o inhabilitados para la recopilación de información de estadísticas de canal.

Puede establecer la recopilación de información de estadísticas de canal en uno de los tres niveles de supervisión: baja, intermedia o alta. Puede establecer el nivel de supervisión a nivel de objeto, o bien a nivel de gestor de colas. La elección de qué nivel utilizar depende del sistema. La recopilación de datos de información de estadísticas puede requerir ciertas instrucciones que resultan relativamente caras a nivel computacional, de forma que, para poder reducir el impacto de la recopilación de información de estadísticas de canal, las opciones de supervisión baja e intermedia miden una muestra de los datos a intervalos regulares, en lugar de recopilar datos todo el tiempo. En la [Tabla 22 en la página 147](#) se resumen los niveles disponibles con la recopilación de información de estadísticas de canal:

<i>Tabla 22. Nivel de detalle de la recopilación de información de estadísticas de canal</i>		
Nivel	Descripción	Utilización
Bajo	Medir una pequeña muestra de los datos, a intervalos regulares.	Para los objetos que procesan un alto volumen de mensajes.

Tabla 22. Nivel de detalle de la recopilación de información de estadísticas de canal (continuación)

Nivel	Descripción	Utilización
Medio	Medir una muestra de los datos, a intervalos regulares.	Para la mayoría de los objetos.
Alto	Medir todos los datos, a intervalos regulares.	Para los objetos que procesan sólo unos pocos mensajes por segundo, en que la información más reciente es importante.

Para cambiar el valor del atributo de canal STATCHL, utilice el mandato MQSC ALTER CHANNEL, y especifique el parámetro STATCHL.

Para cambiar el valor del atributo de gestor de colas STATCHL, utilice el mandato MQSC ALTER QMGR, y especifique el parámetro STATCHL.

Para cambiar el valor del atributo de gestor de colas STATACLS, utilice el mandato MQSC ALTER QMGR, y especifique el parámetro STATACLS.

El atributo de canal, STATCHL, puede tener los valores siguientes:

LOW

La información de estadísticas de canal se recopila con un nivel de detalle bajo.

MEDIUM

La información de estadísticas de canal se recopila con un nivel de detalle intermedio.

HIGH

La información de estadísticas de canal se recopila con un nivel de detalle alto.

OFF

No se recopila información de estadísticas de canal para este canal.

QMGR

El atributo de canal se establece como QMGR. La recopilación de información de estadísticas para este canal se controla mediante el valor del atributo de gestor de colas, STATCHL.

Éste es el valor predeterminado.



Enz/OS sistemas, habilitar este parámetro simplemente activa la recopilación de datos estadísticos, independientemente del valor que seleccione. Si se especifica LOW, MEDIUM o HIGH no hay diferencia en los resultados. Este parámetro debe estar habilitado para poder recopilar los registros de contabilidad de canal.

El atributo de gestor de colas, STATCHL, puede tener los valores siguientes:

LOW

La información de estadísticas de canal se recopila con un nivel de detalle bajo, para todos los canales que tengan el atributo de canal STATCHL establecido como QMGR.

MEDIUM

La información de estadísticas de canal se recopila con un nivel de detalle intermedio, para todos los canales que tengan el atributo de canal STATCHL establecido como QMGR.

HIGH

La información de estadísticas de canal se recopila con un nivel de detalle alto, para todos los canales que tengan el atributo de canal STATCHL establecido como QMGR.


OFF

No se recopila información de estadísticas de cola para todos los canales que tengan el atributo de canal STATCHL establecido como QMGR.

Éste es el valor predeterminado.

NINGUNO

La recopilación de información de estadísticas de canal está inhabilitada para todos los canales, independientemente del atributo de canal STATCHL.

 Enz/OS sistemas, habilitar este parámetro simplemente activa la recopilación de datos estadísticos, independientemente del valor que seleccione. Si se especifica LOW, MEDIUM o HIGH no hay diferencia en los resultados. Este parámetro debe estar habilitado para poder recopilar los registros de contabilidad de canal.

El atributo de gestor de colas, STATACLS, puede tener los valores siguientes:

LOW

La información de las estadísticas se recopila con un nivel de detalle bajo para los canales de clúster emisor definidos automáticamente.

MEDIUM

La información de las estadísticas se recopila con un nivel de detalle intermedio para los canales de clúster emisor definidos automáticamente.

HIGH

La información de las estadísticas se recopila con un nivel de detalle alto para los canales de clúster emisor definidos automáticamente.


OFF

No se recopila información de estadísticas para canales de clúster emisor definidos automáticamente.

QMGR

La recopilación de información de estadísticas para los canales de clúster emisor definidos automáticamente se controla mediante el valor del atributo de gestor de colas, STATCHL.

Éste es el valor predeterminado.

 Enz/OS sistemas, habilitar este parámetro simplemente activa la recopilación de datos estadísticos, independientemente del valor que seleccione. Si se especifica LOW, MEDIUM o HIGH no hay diferencia en los resultados. Este parámetro debe estar habilitado para poder recopilar los registros de contabilidad de canal.

Por ejemplo, para habilitar la recopilación de información de estadísticas, con nivel de detalle intermedio, para el canal emisor QM1 . TO . QM2, utilice el mandato MQSC siguiente:

```
ALTER CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

Para habilitar la recopilación de información de estadísticas, al nivel de detalle intermedio, para todos los canales que especifiquen el atributo de canal STATCHL como QMGR, utilice el mandato MQSC siguiente:

```
ALTER QMGR STATCHL(MEDIUM)
```

Para habilitar la recopilación de información de estadísticas, al nivel de detalle intermedio, para todos los canales de clúster emisor que se definen automáticamente, utilice el mandato MQSC siguiente:

```
ALTER QMGR STATACLS(MEDIUM)
```

Generación de mensajes de estadísticas

Los mensajes de estadísticas se generan a intervalos configurados, y cuando un gestor de colas concluye de forma controlada.

El intervalo configurado se controla mediante el atributo de gestor de colas STATINT, que especifica el intervalo, en segundos, entre la generación de mensajes de estadísticas. El valor del intervalo de estadísticas predeterminado es 1800 segundos (30 minutos). Para cambiar el intervalo de estadísticas,

utilice el mandato MQSC ALTER QMGR, y especifique el parámetro **STATINT**. Por ejemplo, para cambiar el intervalo de estadísticas a 900 segundos (15 minutos) utilice el mandato MQSC siguiente:

```
ALTER QMGR STATINT(900)
```

Para escribir los datos de estadísticas recopilados actualmente en la cola de estadísticas antes de que del intervalo de recopilación de estadísticas caduque, utilice el mandato MQSC RESET QMGR TYPE (STATISTICS). Al emitir este mandato se graban en la cola de estadísticas los datos de estadísticas recopilados, y empieza un nuevo intervalo de recopilación de datos de estadísticas.

Visualización de la información de contabilidad y estadísticas

Para utilizar la información registrada en los mensajes de contabilidad y estadísticas, ejecute una aplicación como, por ejemplo, el programa de ejemplo **amqsmn**, para transformar la información registrada en un formato adecuado

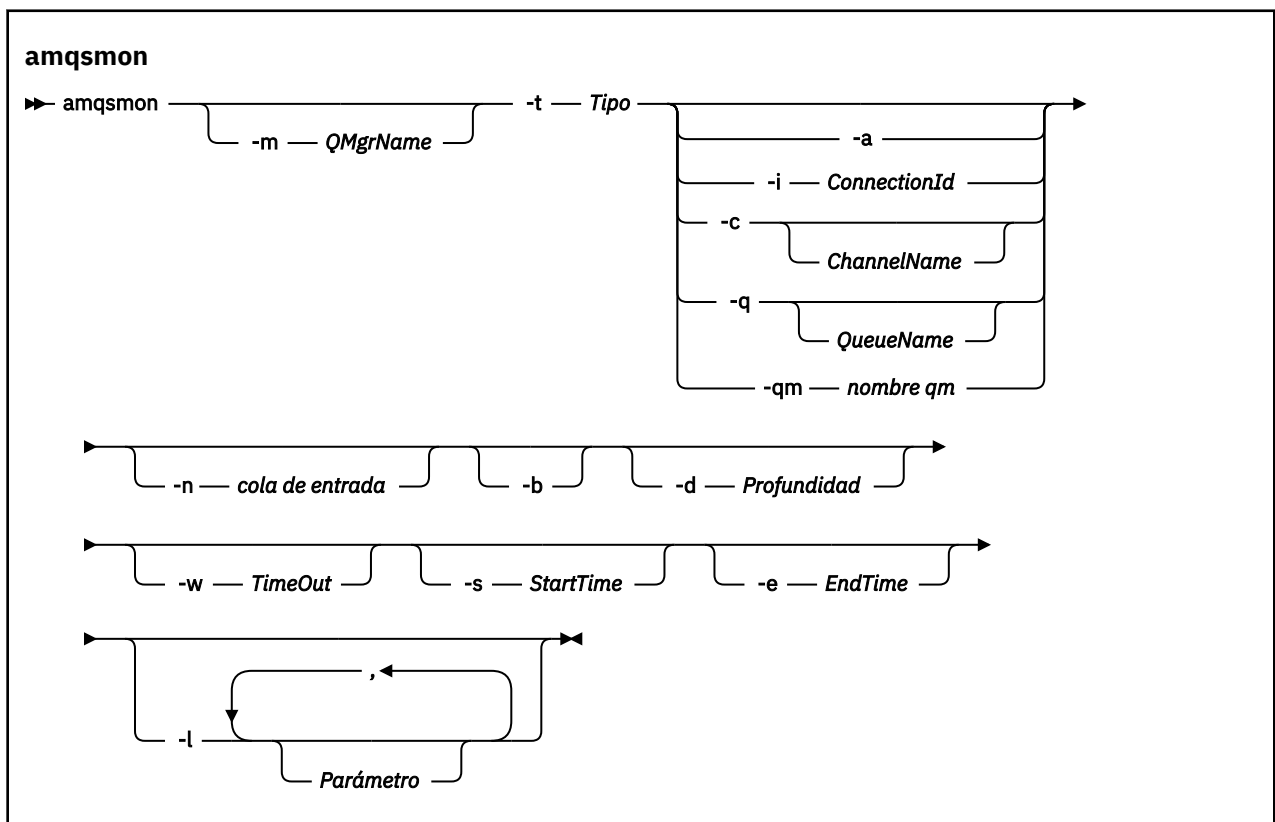
Los mensajes de contabilidad y estadísticas se graban en las colas de contabilidad y estadísticas del sistema. **amqsmn** es un programa de ejemplo que se proporciona con IBM MQ que procesa los mensajes de las colas de contabilidad y estadísticas y muestra la información en la pantalla en un formato legible.

Puesto que **amqsmn** es un programa de ejemplo, puede utilizar el código fuente proporcionado como plantilla para escribir su propia aplicación para procesar mensajes de contabilidad o de estadísticas, o puede modificar el código fuente de **amqsmn** para que satisfaga sus necesidades concretas propias.

amqsmn (Visualizar información de supervisión formateada)

Utilice el programa de ejemplo **amqsmn** para visualizar en un formato legible la información contenida en los mensajes de contabilidad y de estadísticas. El programa **amqsmn** lee los mensajes de contabilidad de la cola de contabilidad, SYSTEM.ADMIN.ACCOUNTING.QUEUE, y lee los mensajes de estadísticas de la cola de estadísticas, SYSTEM.ADMIN.STATISTICS.QUEUE.

Sintaxis



Parámetros necesarios

-t *Tipo*

El tipo de mensajes que se deben procesar. Especifique *Tipo* como uno de los valores siguientes:

contabilidad

Se procesan los registros de contabilidad. Los mensajes se leen de la cola del sistema SYSTEM.ADMIN.ACCOUNTING.QUEUE a menos que haya utilizado el parámetro -n para seleccionar una cola determinada de la que leer.

estadísticas

Se procesan los registros de estadísticas. Los mensajes se leen de la cola del sistema SYSTEM.ADMIN.STATISTICS.QUEUE **V9.4.0** a menos que haya utilizado el parámetro -n para seleccionar una cola determinada de la que leer.

Parámetros opcionales

-m *NombreGestColas*

El nombre del gestor de colas desde el que se procesarán a los mensajes de contabilidad o estadísticas.

Si no especifica este parámetro, se utiliza el gestor de colas predeterminado.

-a

Procesar sólo mensajes que contengan registros MQI.

Mostrar sólo registros MQI. Los mensajes que no contenga registros MQI se dejarán siempre en la cola de la que se hayan leído.

-q *QueueName*

QueueName es un parámetro opcional.

Si no se proporciona *QueueName*: Sólo se visualizan los registros de contabilidad y de estadísticas de la cola.

If *QueueName* is supplied: Sólo se visualizan los registros de contabilidad y estadísticas de cola, de la cola que especifica *QueueName*.

Si no se especifica *-b*, se descartan los mensajes de contabilidad y de estadísticas de los cuales proceden los registros. Puesto que los mensajes de contabilidad y estadísticas también pueden contener registros de otras colas, si no se especifica *-b*, los registros no vistos se pueden descartar.

-c *ChannelName*

ChannelName es un parámetro opcional.

Si no se proporciona *ChannelName*: Sólo se visualizan los registros de estadísticas de canal.

Si se proporciona *NombreCanal*: Sólo se visualizan los registros de estadísticas de canal para el canal que especifica *ChannelName*.

Si no se especifica *-b*, se descartan los mensajes de estadísticas de los cuales proceden los registros. Puesto que los mensajes de estadísticas también puede contener registros de otros canales, si no se especifica *-b*, se pueden descartar los registros que no se han visto.

Este parámetro sólo está disponible al visualizar mensajes de estadísticas, (*-t statistics*).

V 9.4.0**-qm nombre qm**

Este parámetro es opcional, pero si lo especifica, debe proporcionar un nombre de gestor de colas. Especifica que sólo deben leerse de la cola los mensajes de contabilidad o estadísticas producidos por el gestor de colas determinado. Normalmente, sólo se utiliza si los mensajes de varios gestores de colas se han reenviado a un gestor de colas central.

-i ConnectionId

Sólo se visualizan los registros relacionados con el identificador de conexión que especifica *ID_conexión*.

Este parámetro sólo está disponible al visualizar mensajes de contabilidad, (*-t accounting*).

Si no se especifica *-b*, se descartan los mensajes de estadísticas de los cuales proceden los registros. Puesto que los mensajes de estadísticas también puede contener registros de otros canales, si no se especifica *-b*, se pueden descartar los registros que no se han visto.

V 9.4.0**-n cola de entrada**

La cola desde la que se leen los mensajes de contabilidad o estadísticas. Si no se especifica, se utiliza la cola predeterminada para el *-t <tipo>* elegido.

-b

Examinar mensajes.

Los mensajes se recuperan de forma no destructiva.

-d Depth

El número máximo de mensajes que se pueden procesar.

Si no especifica este parámetro, entonces se puede procesar un número ilimitado de mensajes.

-w TimeOut

El número máximo de segundos que se debe esperar a que un mensaje esté disponible.

Si no especifica este parámetro, amqsmn finalizará una vez que no haya más mensajes que procesar.

-s StartTime

Procesar sólo los mensajes transferidos después del valor *HoraInicial* especificado.

StartTime se especifica con el formato *yyyy-mm-dd hh.mm.ss*. Si se especifica una fecha sin indicar la hora, ésta tomará como valor predeterminado *00.00.00*, en la fecha especificada. Las horas se expresan en horario GMT.

Para saber qué efecto tiene no especificar este parámetro, consulte la [Nota 1](#).

-e EndTime

Procesar sólo los mensajes transferidos antes del valor *HoraFinal* especificado.

EndTime se especifica en el formato *yyyy-mm-dd hh.mm.ss*. Si se especifica una fecha sin indicar la hora, ésta tomará como valor predeterminado *23.59.59*, en la fecha especificada. Las horas se expresan en horario GMT.

Para saber qué efecto tiene no especificar este parámetro, consulte la [Nota 1](#).

-l Parámetro

Visualizar sólo los campos seleccionados de los registros procesados. El valor *Parámetro* es una lista de valores enteros separados por comas, en la que cada valor entero se correlaciona con la constante numérica de un campo; consulte el [Ejemplo 5 de amqsmn](#).

Si no especifica este parámetro, se visualizarán todos los campos disponibles.

Nota:

1. Si no especifica *-s HoraInicial* o *-e HoraFinal*, los mensajes que se pueden procesar no están limitados por la hora de transferencia.

Ejemplos de amqsmn (Visualizar información de supervisión formateada)

Utilice esta página para ver ejemplos de la ejecución del programa de ejemplo amqsmn (Visualizar información de supervisión formateada).

1. Consulte “[Datos de mensajes de estadísticas de cola](#)” en la página 194 para obtener una explicación de los atributos.

El mandato siguiente muestra todos los mensajes de estadísticas de MQI del gestor de colas saturn.queue.manager:

```
amqsmn -m saturn.queue.manager -t statistics -a
```

La salida del mandato es la siguiente:

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0
```

2. El mandato siguiente muestra todos los mensajes de estadísticas de cola para la cola LOCALQ en el gestor de colas saturn.queue.manager:

```
amqsmn -m saturn.queue.manager -t statistics -q LOCALQ
```

La salida del mandato es la siguiente:

```
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics:
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  QueueDefinitionType: Local
```

```

QMinDepth: 0
QMaxDepth: 18
AverageQueueTime: [29827281, 0]
PutCount: [26, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [88, 0]
GetCount: [18, 0]
GetBytes: [52, 0]
GetFailCount: 0
BrowseCount: [0, 0]
BrowseBytes: [0, 0]
BrowseFailCount: 1
NonQueuedMsgCount: 0
ExpiredMsgCount: 0
PurgedMsgCount: 0

```

3. El mandato siguiente muestra todos los mensajes de estadísticas registrados desde las 15:30 del 30 de abril de 2005 desde el gestor de colas saturn.queue.manager.

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

La salida del mandato es la siguiente:

```

RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0]
...
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics: 0
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...
QueueStatistics: 1
  QueueName: 'SAMPLEQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...

```

4. Consulte [“Datos de mensajes de contabilidad de cola”](#) en la página 171 para obtener una explicación de los atributos.

El mandato siguiente muestra todos los mensajes de contabilidad registrados el 30 de abril de 2005 desde el gestor de colas saturn.queue.manager:

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

La salida del mandato es la siguiente:

```

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'

```

```

IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
SeqNumber: 0
ApplicationName: 'runmqsc'
ApplicationPid: 8615
ApplicationTid: 1
...

```

5. El mandato siguiente examina la cola de contabilidad y muestra el nombre de la aplicación y el identificador de conexión de todas las aplicaciones para las que está disponible la información de contabilidad MQI:

```
amqsmon -m saturn.queue.manager -t accounting -b -l 7006,3024
```

La salida del mandato es la siguiente:

```

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e80322'
ApplicationName: 'WebSphere MQ\bin\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ea0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting

```

```

ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01e90322'
ApplicationName: 's\IBM\MQ_4\bin64\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01ef0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020f0cf1855e01e60322'
ApplicationName: 's\IBM\MQ_4\bin64\runmqsc.exe'
QueueAccounting: 0

5 Records Processed

```

Consulte [“Búsqueda de la correlación para ApplicationName \(3024\) y ConnectionId \(7006\)”](#) en la página 156 para obtener detalles sobre cómo localizar las variables utilizadas en este tema.

Consulte [“Búsqueda de la correlación para AvgTimeOnQ \(703\) y QmaxDepth \(739\)”](#) en la página 156 para obtener detalles sobre cómo localizar variables si ha seleccionado *statistics* para el parámetro **Type** en el mandato **amqsmn**.

*Búsqueda de la correlación para **ApplicationName** (3024) y **ConnectionId** (7006)*

Utilice este tema para explicar cómo se encuentran las variables utilizadas en el ejemplo, para supervisar datos de contabilidad.

Resumen:

- 3024 significa MQCACF_APPL_NAME que es el **ApplicationName**
- 7006 significa MQBACF_CONNECTION_ID que es el **ConnectionId**

Para averiguar la correspondencia que necesita para llevar a cabo dos procedimientos:

1. Visite [“Datos de mensajes de contabilidad de cola”](#) en la página 171 para obtener una explicación de los atributos [“ApplicationName”](#) en la página 173 y [“ConnectionId”](#) en la página 173 cuando haya seleccionado *contabilidad* para el parámetro **Type** en el mandato **amqsmn**.

En cada caso, busque el atributo **Identifier**.

El **Identifier** para **ApplicationName** es MQCACF_APPL_NAME, y para **ConnectionId** es MQBACF_CONNECTION_ID

2. Busque los identificadores que ha encontrado en el paso “1” en la página 156.

Vaya a la sección [Constants](#) y desplácese hacia abajo hasta que encuentre la lista [MQCACF_*](#) (Tipos de parámetros de caracteres de formato de mandato). Localice MQCACF_APPL_NAME y verá el valor 3024.

De forma similar, busque la lista [MQBACF_*](#) (Tipos de parámetros de byte de formato de mandato). Localice MQBACF_CONNECTION_ID y verá el valor 7006.

Búsqueda de la correlación para AvgTimeOnQ (703) y QmaxDepth (739)

Utilice este tema para explicar cómo se encuentran las variables utilizadas en el ejemplo para supervisar datos de estadísticas.

Resumen:

- 703 significa MQIAMO64_AVG_Q_TIME que es el **AvgTimeOnQ**
- 739 significa MQIAMO_Q_MAX_DEPTH que es el **QMaxDepth**

Para averiguar la correspondencia que necesita para llevar a cabo dos procedimientos:

1. Visite [“Datos de mensajes de estadísticas de cola”](#) en la página 194 para obtener una explicación de los atributos [“AvgTimeOnQ”](#) en la página 198 y [“QMaxDepth”](#) en la página 197 cuando haya seleccionado *statistics* para el parámetro **Type** en el mandato **amqsmn**.

En cada caso, busque el atributo **Identifier**.

Identifier para **AvgTimeOnQ** es MQIAMO64_AVG_Q_TIME y para **QMaxDepth** es MQIAMO_Q_MAX_DEPTH.

2. Busque los identificadores que ha encontrado en el paso [“1”](#) en la página 157.

Vaya a la sección [Constants](#) y desplácese hacia abajo hasta que encuentre la lista [MQIAMO_*](#) (Tipos de parámetros de supervisión de Integer de formato de mandato). Localice MQIAM_Q_MAX_DEPTH y verá el valor 739.

De forma similar, busque la lista [MQIAMO64_*](#) (Formato de mandato 64-bit Integer Monitoring Parameter Types). Localice MQIAMO64_AVG_Q_TIME y verá el valor 703.

Referencia de mensajes contabilidad y estadísticas

Utilice esta página para obtener una visión general del formato de los mensajes de contabilidad y estadísticas, y la información que se devuelve en estos mensajes.

Los mensajes de contabilidad y estadísticas son mensajes estándar de IBM MQ que contienen un descriptor de mensaje y datos de mensaje. Los datos de mensaje contienen información sobre las operaciones MQI realizadas por las aplicaciones IBM MQ, o información sobre las actividades que se producen en un sistema IBM MQ.

Descriptor de mensaje

- Una estructura MQMD

Datos de mensaje

- Una cabecera PCF (MQCFH)
- Los datos de mensajes de contabilidad y mensaje que se devuelven siempre.
- Los datos de mensajes de contabilidad o estadísticas que se devuelven si están disponibles.

Formato de mensajes de contabilidad y estadísticas

Utilice esta página como ejemplo de la estructura de un mensaje de contabilidad MQI.

Tabla 23. Estructura de mensajes de contabilidad MQI

estructura MQMD	Estructura MQCFH de cabecera de mensaje de contabilidad	Datos de mensaje de contabilidad MQI ¹
Identificador de la estructura Versión de la estructura Opciones de informe Tipo de mensaje Hora de caducidad Código de retroalimentación Encoding ID de juego de caracteres codificado Formato de mensaje Prioridad de mensaje Persistence Identificador de mensaje Identificador de correlación Recuento de restituciones Cola de respuestas Gestor de colas de respuestas Identificador de usuario Señal de contabilidad Datos de identidad de la aplicación Tipo de aplicación Nombre de la aplicación Fecha de transferencia Hora de transferencia Datos de origen de la aplicación Identificador de grupo Número de secuencia de mensajes Desplazamiento Distintivos de mensajes Longitud original	Tipo de estructura Longitud de estructura Versión de la estructura Identificador de mandato Número de secuencia de mensajes Opciones de control Código de terminación Código de razón Recuento de parámetros	Gestor de colas Fecha de inicio de intervalo Hora de inicio de intervalo Fecha de final de intervalo Hora de final de intervalo Nivel de mandato Identificador de conexión Número de secuencia Nombre de la aplicación Identificador de proceso de aplicación Identificador de hebra de aplicación Identificador de usuario Fecha de conexión Hora de conexión Nombre de la conexión Nombre de canal Fecha de desconexión Hora de desconexión Tipo de desconexión Recuento de Open Recuento fallido de Open Recuento de Close Recuento fallido de Close Recuento de Put Recuento fallido de Put Recuento de Put1 Recuento fallido de Put1 Bytes de Put Recuento de Get Recuento fallido de Get Bytes de Get Recuento de Browse Recuento fallido de Browse Bytes de Browse Recuento de Commit Recuento fallido de Commit Recuento de restituciones Recuento de Inquire Recuento fallido de Inquire Recuento de Set Recuento fallido de Set
<p>Nota:</p> <p>1. Los parámetros que se muestran son aquellos que se devuelven para un mensaje de contabilidad MQI. Los datos de los mensajes de contabilidad o estadísticas reales dependen de la categoría del mensaje.</p>		

MQMD de mensajes de contabilidad y estadísticas (descriptor de mensaje)

Utilice esta página para comprender las diferencias existentes entre el descriptor de mensaje de los mensajes de contabilidad y de estadísticas y el descriptor de mensaje de los mensajes de suceso.

Los parámetros y valores del descriptor de mensaje de los mensajes de contabilidad y estadísticas son los mismos que los del descriptor de mensaje de los mensajes de suceso, con las excepciones siguientes:

Format

Descripción:	Nombre de formato de los datos del mensaje.
Tipo de datos:	MQCHAR8.
Valor:	MQFMT_ADMIN Mensaje de administración.

Algunos de los parámetros que contiene el descriptor de mensaje de los mensajes de contabilidad y estadísticas contienen datos fijos que proporciona el gestor de colas que ha generado el mensaje.

El MQMD también especifica el nombre del gestor de colas (truncado a los 28 caracteres) que haya transferido la cola de contabilidad o de estadísticas.

Datos de mensajes en mensajes de contabilidad y estadísticas

Los datos de mensaje de los mensajes de contabilidad y estadísticas se basan en el formato de mandato programable (PCF), que se utiliza en las consultas y respuestas de mandato PCF. Los datos de mensaje en los mensajes de contabilidad y estadísticas constan de una cabecera PCF (MQCFH) y de un informe de contabilidad o estadísticas.

MQCFH (cabecera PCF) de mensajes de contabilidad y estadísticas

La cabecera del mensaje de los mensajes de contabilidad y de estadísticas es una estructura MQCFH. Los parámetros y valores de la cabecera del mensaje de los mensajes de contabilidad y estadísticas son los mismos que los de la cabecera del mensaje de los mensajes de suceso, con las excepciones siguientes:

Command

Descripción:	Identificador de mandato. Identifica la categoría de los mensajes de contabilidad o estadísticas.
Tipo de datos:	MQLONG.
Valores:	MQCMD_ACCOUNTING_MQI Mensaje de contabilidad MQI. MQCMD_ACCOUNTING_Q Mensaje de contabilidad de cola. MQCMD_STATISTICS_MQI Mensaje de estadísticas MQI. MQCMD_STATISTICS_Q Mensaje de estadísticas de cola. MQCMD_STATISTICS_CHANNEL Mensaje de estadísticas de canal.

Version

Descripción:	Número de versión de la estructura.
Tipo de datos:	MQLONG.
Valor:	MQCFH_VERSION_3 Version-3 para los mensajes de contabilidad y estadísticas.

Datos de mensajes de contabilidad y estadísticas

El contenido de los datos de mensajes de contabilidad y estadísticas depende de la categoría del mensaje de contabilidad o estadísticas, tal como se indica a continuación:

Mensaje de contabilidad MQI

Los datos de mensaje de contabilidad MQI constan de varios parámetros PCF, pero de ningún grupo PCF.

Mensaje de contabilidad de cola

Los datos del mensaje de contabilidad de cola constan de varios parámetros PCF, y en el rango de 1 a 100 *QAccountingData* grupos PCF.

Mensaje de estadísticas MQI

Los datos de mensaje de estadísticas MQI constan de varios parámetros PCF, pero de ningún grupo PCF.

Mensaje de estadísticas de cola

Los datos de mensaje de estadísticas de cola constan de varios parámetros PCF, y de los grupos PCF *QStatisticsData* que están en el rango de 1 a 100.

Mensaje de estadísticas de canal

Los datos de mensaje de canal constan de varios parámetros PCF, y de los grupos PCF *ChlStatisticsData* que están en el rango de 1 a 100.

datos de mensaje de contabilidad de MQI

Utilice esta página para ver la estructura de un mensaje de contabilidad de MQI.

Nombre de mensaje:	Mensaje de contabilidad MQI.
Plataformas:	Todas, excepto IBM MQ for z/OS.
Cola de sistema:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

Descripción:	El nombre del gestor de colas
Identificador:	MQCA_Q_MGR_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH
Se devuelve:	Siempre

IntervalStartDate

Descripción:	Fecha de inicio del periodo de supervisión
Identificador:	MQCAMO_START_DATE
Tipo de datos:	MQCFST
Longitud máxima:	MQ_DATE_LENGTH
Se devuelve:	Siempre

IntervalStartTime

Descripción:	Hora de inicio del periodo de supervisión
Identificador:	MQCAMO_START_TIME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_TIME_LENGTH
Se devuelve:	Siempre

IntervalEndDate

Descripción:	Fecha de finalización del periodo de supervisión
Identificador:	MQCAMO_END_DATE
Tipo de datos:	MQCFST
Longitud máxima:	MQ_DATE_LENGTH
Se devuelve:	Siempre

IntervalEndTime

Descripción:	Hora de finalización del periodo de supervisión
Identificador:	MQCAMO_END_TIME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_TIME_LENGTH
Se devuelve:	Siempre

CommandLevel

Descripción:	El nivel de mandato del gestor de colas
Identificador:	MQIA_COMMAND_LEVEL
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

ConnectionId

Descripción:	El identificador de conexión para la conexión de IBM MQ
Identificador:	MQBACF_CONNECTION_ID
Tipo de datos:	MQCFBS
Longitud máxima:	MQ_CONNECTION_ID_LENGTH
Se devuelve:	Siempre

SeqNumber

Descripción:	El número de secuencia. Este valor se incrementa para cada registro subsiguiente para las conexiones de larga ejecución.
Identificador:	MQIACF_SEQUENCE_NUMBER
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

ApplicationName

Descripción:	El nombre de la aplicación. El contenido de este campo equivale al contenido del campo <i>PutApplName</i> del descriptor de mensaje.
Identificador:	MQCACF_APPL_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_APPL_NAME_LENGTH
Se devuelve:	Siempre

ApplicationPid

Descripción:	El identificador del proceso del sistema operativo de la aplicación
Identificador:	MQIACF_PROCESS_ID
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

ApplicationTid

Descripción:	El identificador de hebra de IBM MQ de la conexión en la aplicación
Identificador:	MQIACF_THREAD_ID
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

UserId

Descripción:	Contexto de identificador de usuario de la aplicación.
Identificador:	MQCACF_USER_IDENTIFIER
Tipo de datos:	MQCFST
Longitud máxima:	MQ_USER_ID_LENGTH
Se devuelve:	Siempre

ConnDate

Descripción:	Fecha de la operación MQCONN
Identificador:	MQCAMO_CONN_DATE
Tipo de datos:	MQCFST
Longitud máxima:	MQ_TIME_LENGTH
Se devuelve:	Cuando esté disponible

ConnTime

Descripción:	Hora de la operación MQCONN
Identificador:	MQCAMO_CONN_TIME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_TIME_LENGTH
Se devuelve:	Cuando esté disponible

ConnName

Descripción:	Nombre de conexión de la conexión del cliente
Identificador:	MQCACH_CONNECTION_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_CONN_NAME_LENGTH
Se devuelve:	Cuando esté disponible

ChannelName

Descripción:	Nombre de canal de la conexión del cliente
Identificador:	MQCACH_CHANNEL_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_CHANNEL_NAME_LENGTH
Se devuelve:	Cuando esté disponible

RemoteProduct

Descripción:	El identificador de producto remoto para la conexión de cliente como se muestra en el campo RPRODUCT de <u>DISPLAY CHSATUS</u>
Identificador:	MQCACH_REMOTE_PRODUCT
Tipo de datos:	MQCFST
Longitud máxima:	MQ_REMOTE_PRODUCT_LENGTH
Se devuelve:	Cuando esté disponible

RemoteVersion

Descripción:	La versión remota del producto para la conexión de cliente como se muestra en el campo RVERSION de <u>DISPLAY CHSTATUS</u>
Identificador:	MQCACH_REMOTE_VERSION
Tipo de datos:	MQCFST
Longitud máxima:	MQ_REMOTE_VERSION_LENGTH
Se devuelve:	Cuando esté disponible

DiscDate

Descripción:	Fecha de la operación MQDISC
Identificador:	MQCAMO_DISC_DATE
Tipo de datos:	MQCFST
Longitud máxima:	MQ_DATE_LENGTH
Se devuelve:	Cuando esté disponible

DiscTime

Descripción:	Hora de la operación MQDISC
Identificador:	MQCAMO_DISC_TIME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_TIME_LENGTH
Se devuelve:	Cuando esté disponible

DiscType

Descripción:	Tipo de desconexión
Identificador:	MQIAMO_DISC_TYPE
Tipo de datos:	MQCFIN

Valores: Los valores posibles son:
MQDISCONNECT_NORMAL
Solicitado por la aplicación
MQDISCONNECT_IMPLICIT
Terminación de aplicación anómala
MQDISCONNECT_Q_MGR
Conexión interrumpida por el gestor de colas

Se devuelve: Cuando esté disponible

OpenCount

Descripción: El número de objetos abiertos satisfactoriamente, ya sea emitiendo directamente una llamada a MQOPEN o utilizando el verbo MQPUT1. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).

Identificador: MQIAMO_OPENS

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

OpenFailCount

Descripción: El número de intentos de abrir un objeto no satisfactorios. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).

Identificador: MQIAMO_OPENS_FAILED

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

CloseCount

Descripción: El número de objetos cerrados. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).

Identificador: MQIAMO_CLOSES

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

CloseFailCount

Descripción: El número de intentos de cerrar un objeto no satisfactorios. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).

Identificador: MQIAMO_CLOSES_FAILED

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

PutCount

Descripción: El número de mensajes persistentes y no persistentes transferidos satisfactoriamente a una cola, con la excepción de los mensajes transferidos utilizando la llamada MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO_PUTS
Tipo de datos: MQCFIL
Se devuelve: Cuando esté disponible

PutFailCount

Descripción: El número de intentos de colocación de un mensaje no satisfactorios
Identificador: MQIAMO_PUTS_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible

Put1Count

Descripción: El número de mensajes persistentes y no persistentes colocados satisfactoriamente en la cola mediante llamadas MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_PUT1S
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

Put1FailCount

Descripción: El número de intentos de colocación de un mensaje mediante llamadas MQPUT1 no satisfactorios
Identificador: MQIAMO_PUT1S_FAILED
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

PutBytes

Descripción: El número de bytes escritos utilizando llamadas de colocación para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_PUT_BYTES
Tipo de datos: MQCFIL64
Se devuelve: Cuando esté disponible

GetCount

Descripción: El número de llamadas MQGET destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_GETS
Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

GetFailCount

Descripción: El número de llamadas MQGET destructivas que han fallado

Identificador: MQIAMO_GETS_FAILED

Tipo de datos: MQCFIN

Se devuelve: Cuando esté disponible

GetBytes

Descripción: Número total de bytes recuperados para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO64_GET_BYTES

Tipo de datos: MQCFIL64

Se devuelve: Cuando esté disponible

BrowseCount

Descripción: El número de llamadas MQGET no destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO_BROWSES

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

BrowseFailCount

Descripción: El número de llamadas MQGET no destructivas no satisfactorias

Identificador: MQIAMO_BROWSES_FAILED

Tipo de datos: MQCFIN

Se devuelve: Cuando esté disponible

BrowseBytes

Descripción: Número total de bytes examinados para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO64_BROWSE_BYTES

Tipo de datos: MQCFIL64

Se devuelve: Cuando esté disponible

CommitCount

Descripción: El número de transacciones satisfactorias. Este número incluye aquellas transacciones que la aplicación conectada confirma implícitamente. En esta cuenta se incluyen las solicitudes de confirmación en las que no hay ningún trabajo pendiente.

Identificador: MQIAMO_COMMITS

Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible

CommitFailCount

Descripción: El número de intentos de completar una transacción no satisfactorios
Identificador: MQIAMO_COMMITS_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible

BackCount

Descripción: El número de restituciones procesadas, incluidas las restituciones implícitas debido a una desconexión anormal
Identificador: MQIAMO_BACKOUTS
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible

InqCount

Descripción: El número de objetos cuya consulta ha resultado satisfactoria. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_INQS
Tipo de datos: MQCFIL
Se devuelve: Cuando esté disponible

InqFailCount

Descripción: El número de intentos de consulta de objetos no satisfactorios. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_INQS_FAILED
Tipo de datos: MQCFIL
Se devuelve: Cuando esté disponible

SetCount

Descripción: El número de llamadas MQSET satisfactorias. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_SETS
Tipo de datos: MQCFIL
Se devuelve: Cuando esté disponible

SetFailCount

Descripción: El número de llamadas MQSET no satisfactorias. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_SETS_FAILED
Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible

SubCountDur

Descripción: El número de solicitudes de suscripción satisfactorias que han creado, alterado o reanudado suscripciones duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 = El número de suscripciones creadas

1 = El número de suscripciones alteradas

2 = El número de suscripciones reanudadas

Identificador: MQIAMO_SUBS_DUR

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible.

SubCountNDur

Descripción: El número de solicitudes de suscripción satisfactorias que han creado, alterado o reanudado suscripciones no duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 = El número de suscripciones creadas

1 = El número de suscripciones alteradas

2 = El número de suscripciones reanudadas

Identificador: MQIAMO_SUBS_NDUR

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible.

SubFailCount

Descripción: El número de solicitudes de suscripción satisfactorias.

Identificador: MQIAMO_SUBS_FAILED

Tipo de datos: MQCFIN

Se devuelve: Cuando esté disponible.

UnsubCountDur

Descripción: El número de solicitudes de cancelación de suscripción satisfactorias para las suscripciones duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 - La suscripción se ha cerrado pero no se ha eliminado

1 - La suscripción se ha cerrado y eliminado

Identificador: MQIAMO_UNSUBS_DUR

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible.

UnsubCountNDur

Descripción: El número de solicitudes de cancelación de suscripción satisfactorias para las suscripciones duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 - La suscripción se ha cerrado pero no se ha eliminado
1 - La suscripción se ha cerrado y eliminado

Identificador: MQIAMO_UNSUBS_NDUR
Tipo de datos: MQCFIL
Se devuelve: Cuando esté disponible.

UnsubFailCount

Descripción: El número de solicitudes de anulación de suscripción no satisfactorias.

Identificador: MQIAMO_UNSUBS_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

SubRqCount

Descripción: El número de solicitudes MQSUBRQ satisfactorias.

Identificador: MQIAMO_SUBRQS
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

SubRqFailCount

Descripción: El número de solicitudes MQSUB no satisfactorias.

Identificador: MQIAMO_SUBRQS_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

CBCount

Descripción: El número de solicitudes MQCB satisfactorias. Esta es una matriz de valores indexada por el tipo de operación.

0 - Una devolución de llamada se ha creado o alterado
1 - Se ha eliminado una devolución de llamada
2 - Se ha reanudado una devolución de llamada
3 - Se ha suspendido una devolución de llamada

Identificador: MQIAMO_CBS
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

CBFailCount

Descripción: El número de solicitudes MQCB satisfactorias.

Identificador: MQIAMO_CBS_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

CtlCount

Descripción: El número de solicitudes MQCTL satisfactorias. Esta es una matriz de valores indexada por el tipo de operación.
0 - La conexión se ha iniciado
1 - Se ha detenido la conexión
2 - La conexión se ha reanudado
3 - Se ha suspendido la conexión

Identificador: MQIAMO_CTLs
Tipo de datos: MQCFIL
Se devuelve: Cuando esté disponible.

CtlFailCount

Descripción: El número de solicitudes QCTL no satisfactorias.
Identificador: MQIAMO_CTLs_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

StatCount

Descripción: El número de solicitudes MQSTAT satisfactorias.
Identificador: MQIAMO_STATS.
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

StatFailCount

Descripción: El número de solicitudes MQSTAT no satisfactorias.
Identificador: MQIAMO_STATS_FAILED
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

PutTopicCount

Descripción: El número de mensajes persistentes y no persistentes transferidos satisfactoriamente a un tema, con la excepción de los mensajes transferidos utilizando la llamada MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Nota: en este valor se incluyen los mensajes colocados mediante un alias de cola que se resuelven en un tema.

Identificador: MQIAMO_TOPIC_PUTS
Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible.

PutTopicFailCount

Descripción: El número de intentos de colocación de un mensaje en un tema no satisfactorios.

Identificador: MQIAMO_TOPIC_PUTS_FAILED

Tipo de datos: MQCFIN

Se devuelve: Cuando esté disponible.

Put1TopicCount

Descripción: El número de mensajes persistentes y no persistentes colocados satisfactoriamente en un tema mediante llamadas MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Nota: en este valor se incluyen los mensajes colocados mediante un alias de cola que se resuelven en un tema.

Identificador: MQIAMO_TOPIC_PUT1S

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible.

Put1TopicFailCount

Descripción: El número de intentos de colocación de un mensaje en un tema no satisfactorios mediante llamadas MQPUT1.

Identificador: MQIAMO_TOPIC_PUT1S_FAILED

Tipo de datos: MQCFIN

Se devuelve: Cuando esté disponible.

PutTopicBytes

Descripción: El número de bytes escritos utilizando llamadas de colocación para los mensajes persistentes y no persistentes que resuelven en una operación de tipo publish. Éste es el número de bytes que la aplicación coloca, y no el número resultante de bytes entregados a los suscriptores. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO64_TOPIC_PUT_BYTES

Tipo de datos: MQCFIL64

Se devuelve: Cuando esté disponible.

Datos de mensajes de contabilidad de cola

Utilice esta página para ver la estructura de un mensaje de contabilidad de cola.

Nombre de mensaje:	Mensaje de contabilidad de cola.
Plataformas:	Todas, excepto IBM MQ for z/OS.
Cola de sistema:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

Descripción: El nombre del gestor de colas
Identificador: MQCA_Q_MGR_NAME
Tipo de datos: MQCFST
Longitud máxima: MQ_Q_MGR_NAME_LENGTH
Se devuelve: Siempre

IntervalStartDate

Descripción: Fecha de inicio del periodo de supervisión
Identificador: MQCAMO_START_DATE
Tipo de datos: MQCFST
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre

IntervalStartTime

Descripción: Hora de inicio del periodo de supervisión
Identificador: MQCAMO_START_TIME
Tipo de datos: MQCFST
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Siempre

IntervalEndDate

Descripción: Fecha de finalización del periodo de supervisión
Identificador: MQCAMO_END_DATE
Tipo de datos: MQCFST
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre

IntervalEndTime

Descripción: Hora de finalización del periodo de supervisión
Identificador: MQCAMO_END_TIME
Tipo de datos: MQCFST
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Siempre

CommandLevel

Descripción: El nivel de mandato del gestor de colas
Identificador: MQIA_COMMAND_LEVEL
Tipo de datos: MQCFIN
Se devuelve: Siempre

ConnectionId

Descripción:	El identificador de conexión para la conexión de IBM MQ
Identificador:	MQBACF_CONNECTION_ID
Tipo de datos:	MQCFBS
Longitud máxima:	MQ_CONNECTION_ID_LENGTH
Se devuelve:	Siempre

SeqNumber

Descripción:	El número de secuencia. Este valor se incrementa para cada registro subsiguiente para las conexiones de larga ejecución.
Identificador:	MQIACF_SEQUENCE_NUMBER
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

ApplicationName

Descripción:	El nombre de la aplicación. El contenido de este campo equivale al contenido del campo PutApplName del descriptor de mensaje.
Identificador:	MQCACF_APPL_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_APPL_NAME_LENGTH
Se devuelve:	Siempre

ApplicationPid

Descripción:	El identificador del proceso del sistema operativo de la aplicación
Identificador:	MQIACF_PROCESS_ID
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

ApplicationTid

Descripción:	El identificador de hebra de IBM MQ de la conexión en la aplicación
Identificador:	MQIACF_THREAD_ID
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

UserId

Descripción:	Contexto de identificador de usuario de la aplicación.
Identificador:	MQCACF_USER_IDENTIFIER
Tipo de datos:	MQCFST
Longitud máxima:	MQ_USER_ID_LENGTH
Se devuelve:	Siempre

ChannelName

Descripción:	Nombre de canal de la conexión del cliente
Identificador:	MQCACH_CHANNEL_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_CHANNEL_NAME_LENGTH
Se devuelve:	Cuando esté disponible

ConnName

Descripción:	Nombre de conexión de la conexión del cliente
Identificador:	MQCACH_CONNECTION_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_CONN_NAME_LENGTH
Se devuelve:	Cuando esté disponible

ObjectCount

Descripción:	El número de colas a las que se accede en el intervalo para el que se han registrado los datos contables. El valor se establece en el número de grupos PCF <i>QAccountingData</i> que el mensaje contiene.
Identificador:	MQIAMO_OBJECT_COUNT
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

QAccountingData

Descripción:	Parámetros agrupados que especifican los detalles de contabilidad de una cola
Identificador:	MQGACF_Q_ACCOUNTING_DATA
Tipo de datos:	MQCFGR

Parámetros del grupo:	<i>QName</i> <i>CreateDate</i> <i>CreateTime</i> <i>QType</i> <i>QDefinitionType</i> <i>OpenCount</i> <i>OpenDate</i> <i>OpenTime</i> <i>CloseDate</i> <i>CloseTime</i> <i>PutCount</i> <i>PutFailCount</i> <i>Put1Count</i> <i>Put1FailCount</i> <i>PutBytes</i> <i>PutMinBytes</i> <i>PutMaxBytes</i> <i>GetCount</i> <i>GetFailCount</i> <i>GetBytes</i> <i>GetMinBytes</i> <i>GetMaxBytes</i> <i>BrowseCount</i> <i>BrowseFailCount</i> <i>BrowseBytes</i> <i>BrowseMinBytes</i> <i>BrowseMaxBytes</i> <i>TimeOnQMin</i> <i>TimeOnQAvg</i> <i>TimeOnQMax</i>
-----------------------	---

Se devuelve: Siempre

QName

Descripción:	El nombre de la cola
Identificador:	MQCA_Q_NAME
Tipo de datos:	MQCFST
Incluido en el grupo PCF:	<i>QAccountingData</i>
Longitud máxima:	MQ_Q_NAME_LENGTH
Se devuelve:	Cuando esté disponible

CreateDate

Descripción:	La fecha en que se ha creado la cola
Identificador:	MQCA_CREATION_DATE
Tipo de datos:	MQCFST
Incluido en el grupo PCF:	<i>QAccountingData</i>

Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Cuando esté disponible

CreateTime

Descripción: La hora en que se ha creado la cola
Identificador: MQCA_CREATION_TIME
Tipo de datos: MQCFST
Incluido en el grupo PCF: *QAccountingData*
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Cuando esté disponible

QType

Descripción: El tipo de la cola
Identificador: MQIA_Q_TYPE
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QAccountingData*
Valor: MQQT_LOCAL
Se devuelve: Cuando esté disponible

QDefinitionType

Descripción: El tipo de definición de cola
Identificador: MQIA_DEFINITION_TYPE
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QAccountingData*
Valores: Los valores posibles son:
MQQDT_PREDEFINED
MQQDT_PERMANENT_DYNAMIC
MQQDT_TEMPORARY_DYNAMIC
Se devuelve: Cuando esté disponible

OpenCount

Descripción: El número de veces que la aplicación ha abierto esta cola en este intervalo, directamente emitiendo una llamada a MQOPEN o utilizando el verbo MQPUT1.
Identificador: MQIAMO_OPENS
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

OpenDate

Descripción:	La fecha en que la cola se abrió por primera vez en este intervalo de registro. Si la cola ya estaba abierta al inicio de este intervalo, este valor refleja la fecha en que la cola se abrió originariamente.
Identificador:	MQCAMO_OPEN_DATE
Tipo de datos:	MQCFST
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

OpenTime

Descripción:	La hora en que la cola se abrió por primera vez en este intervalo de registro. Si la cola ya estaba abierta al inicio de este intervalo, este valor refleja la hora en que la cola se abrió originariamente.
Identificador:	MQCAMO_OPEN_TIME
Tipo de datos:	MQCFST
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

CloseDate

Descripción:	La fecha del cierre final de la cola en este intervalo de registro. Si la cola aún sigue abierta, no se devuelve el valor.
Identificador:	MQCAMO_CLOSE_DATE
Tipo de datos:	MQCFST
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

CloseTime

Descripción:	La hora del cierre final de la cola en este intervalo de registro. Si la cola aún sigue abierta, no se devuelve el valor.
Identificador:	MQCAMO_CLOSE_TIME
Tipo de datos:	MQCFST
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

PutCount

Descripción:	El número de mensajes persistentes y no persistentes transferidos satisfactoriamente a la cola, con la excepción de las llamadas MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_PUTS
Tipo de datos:	MQCFIL

Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

PutFailCount

Descripción: El número de intentos de colocación de un mensaje no satisfactorios, con la excepción de las llamadas MQPUT1.
Identificador: MQIAMO_PUTS_FAILED
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

Put1Count

Descripción: El número de mensajes persistentes y no persistentes colocados satisfactoriamente en la cola mediante llamadas MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_PUT1S
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

Put1FailCount

Descripción: El número de intentos de colocación de un mensaje mediante llamadas MQPUT1 no satisfactorios
Identificador: MQIAMO_PUT1S_FAILED
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

PutBytes

Descripción: El número total de bytes colocados para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_PUT_BYTES
Tipo de datos: MQCFIL64
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

PutMinBytes

Descripción:	El tamaño de mensaje persistente y no persistente más pequeño colocado en la cola. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_PUT_MIN_BYTES
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

PutMaxBytes

Descripción:	El tamaño de mensaje persistente y no persistente más grande colocado en la cola. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_PUT_MAX_BYTES
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

GeneratedMsgCount

Descripción:	El número de mensajes generados. Los mensajes generados son: <ul style="list-style-type: none">• Sucesos de profundidad de cola alta• Sucesos de profundidad de cola baja
Identificador:	MQIAMO_GENERATED_MSGS
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

GetCount

Descripción:	El número de llamadas MQGET destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_GETS
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

GetFailCount

Descripción:	El número de llamadas MQGET destructivas que han fallado
Identificador:	MQIAMO_GETS_FAILED
Tipo de datos:	MQCFIN

Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

GetBytes

Descripción: El número de bytes leídos en las colas MQGET destructivas para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_GET_BYTES
Tipo de datos: MQCFIL64
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

GetMinBytes

Descripción: El tamaño de mensaje persistente y no persistente más pequeño recuperado de la cola. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_GET_MIN_BYTES
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

GetMaxBytes

Descripción: El tamaño de mensaje persistente y no persistente más grande recuperado de la cola. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_GET_MAX_BYTES
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

BrowseCount

Descripción: El número de llamadas MQGET no destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_BROWSES
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QAccountingData*
Se devuelve: Cuando esté disponible

BrowseFailCount

Descripción:	El número de llamadas MQGET no destructivas no satisfactorias
Identificador:	MQIAMO_BROWSES_FAILED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

BrowseBytes

Descripción:	El número de bytes leídos en las llamadas MQGET no destructivas que los mensajes persistentes han devuelto
Identificador:	MQIAMO64_BROWSE_BYTES
Tipo de datos:	MQCFIL64
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

BrowseMinBytes

Descripción:	El tamaño de mensaje persistente y no persistente más pequeño examinado en la cola. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_BROWSE_MIN_BYTES
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

BrowseMaxBytes

Descripción:	El tamaño de mensaje persistente y no persistente más grande examinado en la cola. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_BROWSE_MAX_BYTES
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QAccountingData</i>
Se devuelve:	Cuando esté disponible

TimeOnQMin

Descripción:	El espacio de tiempo más corto que un mensaje persistente y no persistente ha permanecido en la cola antes de que se haya recuperado de forma destructiva, en microsegundos. En el caso de los mensajes recuperados bajo el punto de sincronismo, este valor no incluye el tiempo transcurrido antes de que se haya confirmado la operación get. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO64_Q_TIME_MIN

Tipo de datos: MQCFIL64
Incluido en el grupo PCF: QAccountingData
Se devuelve: Cuando esté disponible

TimeOnQAvg

Descripción: El promedio de tiempo que un mensaje persistente y no persistente ha permanecido en la cola antes de que se haya recuperado de forma destructiva, en microsegundos. En el caso de los mensajes recuperados bajo el punto de sincronismo, este valor no incluye el tiempo transcurrido antes de que se haya confirmado la operación get. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO64_Q_TIME_AVG
Tipo de datos: MQCFIL64
Incluido en el grupo PCF: QAccountingData
Se devuelve: Cuando esté disponible

TimeOnQMax

Descripción: El espacio de tiempo más largo que un mensaje persistente y no persistente ha permanecido en la cola antes de que se haya recuperado de forma destructiva, en microsegundos. En el caso de los mensajes recuperados bajo el punto de sincronismo, este valor no incluye el tiempo transcurrido antes de que se haya confirmado la operación get. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Identificador: MQIAMO64_Q_TIME_MAX
Tipo de datos: MQCFIL64
Incluido en el grupo PCF: QAccountingData
Se devuelve: Cuando esté disponible

CBCount

Descripción: El número de solicitudes MQCB satisfactorias. Esta es una matriz de valores indexada por el tipo de operación.

- 0 - Una devolución de llamada se ha creado o alterado
- 1 - Se ha eliminado una devolución de llamada
- 2 - Se ha reanudado una devolución de llamada
- 3 - Se ha suspendido una devolución de llamada

Identificador: MQIAMO_CBS
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

CBFailCount

Descripción: El número de solicitudes MQCB satisfactorias.

Identificador: MQIAMO_CBS_FAILED

Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

datos de mensaje de estadísticas de MQI

Utilice esta página para ver la estructura de un mensaje de estadísticas MQI.

Nombre de mensaje:	Mensaje de estadísticas MQI.
Plataformas:	Todas, excepto IBM MQ for z/OS.
Cola de sistema:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Descripción: Nombre del gestor de colas.
Identificador: MQCA_Q_MGR_NAME.
Tipo de datos: MQCFST.
Longitud máxima: MQ_Q_MGR_NAME_LENGTH.
Se devuelve: Siempre.

IntervalStartDate

Descripción: La fecha del inicio del periodo de supervisión.
Identificador: MQCAMO_START_DATE.
Tipo de datos: MQCFST.
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre.

IntervalStartTime

Descripción: La hora del inicio del periodo de supervisión.
Identificador: MQCAMO_START_TIME.
Tipo de datos: MQCFST.
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Siempre.

IntervalEndDate

Descripción: La fecha del final del periodo de supervisión.
Identificador: MQCAMO_END_DATE.
Tipo de datos: MQCFST.
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre.

IntervalEndTime

Descripción: La hora del final del periodo de supervisión.
Identificador: MQCAMO_END_TIME.
Tipo de datos: MQCFST.

Longitud máxima: MQ_TIME_LENGTH

Se devuelve: Siempre.

CommandLevel

Descripción: El nivel de mandato del gestor de colas.

Identificador: MQIA_COMMAND_LEVEL.

Tipo de datos: MQCFIN.

Se devuelve: Siempre.

ConnCount

Descripción: El número de conexiones satisfactorias con el gestor de colas.

Identificador: MQIAMO_CONNS.

Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

ConnFailCount

Descripción: El número de intentos de conexión no satisfactorios.

Identificador: MQIAMO_CONNS_FAILED.

Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

ConnsMax

Descripción: El número máximo de conexiones simultáneas en el intervalo de registro.

Identificador: MQIAMO_CONNS_MAX.

Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

DiscCount

Descripción: El número de desconexiones del gestor de colas. Se trata de una matriz de enteros, indexada por las constantes siguientes:

- MQDISCONNECT_NORMAL
- MQDISCONNECT_IMPLICIT
- MQDISCONNECT_Q_MGR

Identificador: MQIAMO_DISCS.

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

OpenCount

Descripción: El número de objetos abiertos satisfactoriamente, ya sea emitiendo directamente una llamada a MQOPEN o utilizando el verbo MQPUT1. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).

Identificador: MQIAMO_OPENS.

Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

OpenFailCount

Descripción: El número de intentos no satisfactorios de abrir un objeto. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_OPENS_FAILED.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

CloseCount

Descripción: El número de objetos que se han cerrado de forma satisfactoria. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_CLOSES.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

CloseFailCount

Descripción: El número de intentos no satisfactorios de cerrar un objeto. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_CLOSES_FAILED.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

InqCount

Descripción: El número de objetos cuya consulta ha resultado satisfactoria. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_INQS.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

InqFailCount

Descripción: El número de intentos de consulta de objetos no satisfactorios. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la [Nota de referencia 1](#).
Identificador: MQIAMO_INQS_FAILED.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

SetCount

Descripción:	El número de objetos actualizados de forma satisfactoria (SET). Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la Nota de referencia 1 .
Identificador:	MQIAMO_SETS.
Tipo de datos:	MQCFIL.
Se devuelve:	Cuando esté disponible.

SetFailCount

Descripción:	El número de intentos de tipo SET no satisfactorios. Este parámetro es una lista de enteros indexada por tipo de objeto; consulte la Nota de referencia 1 .
Identificador:	MQIAMO_SETS_FAILED.
Tipo de datos:	MQCFIL.
Se devuelve:	Cuando esté disponible.

PutCount

Descripción:	El número de mensajes persistentes y no persistentes transferidos satisfactoriamente a una cola, con la excepción de las solicitudes MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la Nota de referencia 2 .
Identificador:	MQIAMO_PUTS.
Tipo de datos:	MQCFIL.
Se devuelve:	Cuando esté disponible.

PutFailCount

Descripción:	El número de intentos no satisfactorios de transferencia de mensaje.
Identificador:	MQIAMO_PUTS_FAILED.
Tipo de datos:	MQCFIN.
Se devuelve:	Cuando esté disponible.

Put1Count

Descripción:	El número de mensajes persistentes y no persistentes colocados satisfactoriamente en una cola mediante solicitudes MQPUT1. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte Nota de referencia 2 .
Identificador:	MQIAMO_PUT1S.
Tipo de datos:	MQCFIL.
Se devuelve:	Cuando esté disponible.

Put1FailCount

Descripción:	El número de intentos no satisfactorios de colocación de un mensaje persistente y no persistente en una cola mediante solicitudes MQPUT1. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte Nota de referencia 2 .
Identificador:	MQIAMO_PUT1S_FAILED.

Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

PutBytes

Descripción: El número de bytes de los mensajes persistentes y no persistentes escritos mediante las solicitudes de tipo put. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte [Nota de referencia 2](#).
Identificador: MQIAMO64_PUT_BYTES.
Tipo de datos: MQCFIL64.
Se devuelve: Cuando esté disponible.

GetCount

Descripción: El número de solicitudes de obtención destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte [Nota de referencia 2](#).
Identificador: MQIAMO_GETS.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

GetFailCount

Descripción: El número de solicitudes de obtención destructivas no satisfactorias.
Identificador: MQIAMO_GETS_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

GetBytes

Descripción: El número de bytes leídos en las solicitudes de obtención destructivas para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte [Nota de referencia 2](#).
Identificador: MQIAMO64_GET_BYTES.
Tipo de datos: MQCFIL64.
Se devuelve: Cuando esté disponible.

BrowseCount

Descripción: El número de solicitudes de obtención no destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte [Nota de referencia 2](#).
Identificador: MQIAMO_BROWSES.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

BrowseFailCount

Descripción: El número de solicitudes de obtención no destructivas no satisfactorias.
Identificador: MQIAMO_BROWSES_FAILED.

Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

BrowseBytes

Descripción: El número de bytes leídos en las solicitudes de obtención no destructivas para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexados por valor de persistencia; consulte [Nota de referencia 2](#).
Identificador: MQIAMO64_BROWSE_BYTES.
Tipo de datos: MQCFIL64.
Se devuelve: Cuando esté disponible.

CommitCount

Descripción: El número de transacciones que se ha completado de forma satisfactoria. Este número incluye las transacciones confirmadas implícitamente por la desconexión de la aplicación, y confirma las solicitudes en las que no haya trabajos pendientes.
Identificador: MQIAMO_COMMITS.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

CommitFailCount

Descripción: El número de intentos de completar una transacción no satisfactorios.
Identificador: MQIAMO_COMMITS_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

BackCount

Descripción: El número de restituciones procesadas, incluidas las restituciones implícitas debido a una desconexión anormal.
Identificador: MQIAMO_BACKOUTS.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

ExpiredMsgCount

Descripción: El número de mensajes persistentes y no persistentes que se han descartado porque habían caducado antes de que se pudieran recuperar.
Identificador: MQIAMO_MSGS_EXPIRED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

PurgeCount

Descripción: El número de veces que se ha borrado la cola.
Identificador: MQIAMO_MSGS_PURGED.
Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

SubCountDur

Descripción: El número de solicitudes de suscripción satisfactorias que han creado, modificado o reanudado las suscripciones duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 = El número de suscripciones creadas

1 = El número de suscripciones alteradas

2 = El número de suscripciones reanudadas

Identificador: MQIAMO_SUBS_DUR.

Tipo de datos: MQCFIL

Se devuelve: Cuando esté disponible.

SubCountNDur

Descripción: El número de solicitudes de suscripción satisfactorias que han creado, modificado o reanudado las suscripciones no duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 = El número de suscripciones creadas

1 = El número de suscripciones alteradas

2 = El número de suscripciones reanudadas

Identificador: MQIAMO_SUBS_NDUR.

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

SubFailCount

Descripción: El número de solicitudes de suscripción satisfactorias.

Identificador: MQIAMO_SUBS_FAILED.

Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

UnsubCountDur

Descripción: El número de solicitudes de cancelación de suscripción satisfactorias para las suscripciones duraderas. Esta es una matriz de valores indexada por el tipo de operación.

0 - La suscripción se ha cerrado pero no se ha eliminado

1 - La suscripción se ha cerrado y eliminado

Identificador: MQIAMO_UNSUBS_DUR.

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

UnsubCountNDur

Descripción: El número de solicitudes de anulación de suscripción para las suscripciones no duraderas. Esta es una matriz de valores indexada por el tipo de operación.
0 - La suscripción se ha cerrado pero no se ha eliminado
1 - La suscripción se ha cerrado y eliminado

Identificador: MQIAMO_UNSUBS_NDUR.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

UnsubFailCount

Descripción: El número de solicitudes de anulación de suscripción fallidas.
Identificador: MQIAMO_UNSUBS_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

SubRqCount

Descripción: El número de solicitudes MQSUBRQ satisfactorias.
Identificador: MQIAMO_SUBRQS
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

SubRqFailCount

Descripción: El número de solicitudes MQSUBRQ no satisfactorias.
Identificador: MQIAMO_SUBRQS_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

CBCount

Descripción: El número de solicitudes MQCB satisfactorias. Esta es una matriz de valores indexada por el tipo de operación.
0 - Una devolución de llamada se ha creado o alterado
1 - Se ha eliminado una devolución de llamada
2 - Se ha reanudado una devolución de llamada
3 - Se ha suspendido una devolución de llamada

Identificador: MQIAMO_CBS.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

CBFailCount

Descripción: El número de solicitudes MQCB satisfactorias.
Identificador: MQIAMO_CBS_FAILED.

Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

CtlCount

Descripción: El número de solicitudes MQCTL satisfactorias. Es una matriz de valores indexados por el tipo de operación:

- 0 - La conexión se ha iniciado
- 1 - Se ha detenido la conexión
- 2 - La conexión se ha reanudado
- 3 - Se ha suspendido la conexión

Identificador: MQIAMO_CTLS.
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

CtlFailCount

Descripción: El número de solicitudes QCTL no satisfactorias.

Identificador: MQIAMO_CTLS_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

StatCount

Descripción: El número de solicitudes MQSTAT satisfactorias.

Identificador: MQIAMO_STATS.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

StatFailCount

Descripción: El número de solicitudes MQSTAT no satisfactorias.

Identificador: MQIAMO_STATS_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

SubCountDurHighWater

Descripción: La marca de límite superior del número de suscripciones duraderas durante el intervalo de tiempo. Ésta es una matriz de valores indexada por el valor SUBTYPE.

- 0 - La marca de límite superior para todas las suscripciones duraderas en el sistema
- 1 - La marca de límite superior para las suscripciones de aplicación duraderas (MQSUBTYPE_API)
- 2 - La marca de límite superior para las suscripciones de administración duraderas (MQSUBTYPE_ADMIN)
- 3 - La marca de límite superior para las suscripciones de proxy duraderas (MQSUBTYPE_PROXY)

Identificador: MQIAMO_SUB_DUR_HIGHWATER

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

SubCountDurLowWater

Descripción: La marca de límite inferior del número de suscripciones duraderas durante el intervalo de tiempo. Ésta es una matriz de valores indexada por el valor SUBTYPE.

- 0 - La marca de límite inferior para todas las suscripciones duraderas en el sistema
- 1 - La marca de límite inferior para las suscripciones de aplicación duraderas (MQSUBTYPE_API)
- 2 - La marca de límite inferior para las suscripciones de administración duraderas (MQSUBTYPE_ADMIN)
- 3 - La marca de límite inferior para las suscripciones de proxy duraderas (MQSUBTYPE_PROXY)

Identificador: MQIAMO_SUB_DUR_LOWWATER

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

SubCountNDurHighWater

Descripción: La marca de límite superior del número de suscripciones no duraderas durante el intervalo de tiempo. Ésta es una matriz de valores indexada por el valor SUBTYPE.

- 0 - La marca de límite superior para todas las suscripciones no duraderas en el sistema
- 1 - La marca de límite superior para las suscripciones de aplicación no duraderas (MQSUBTYPE_API)
- 2 - La marca de límite superior para las suscripciones de administración no duraderas (MQSUBTYPE_ADMIN)
- 3 - La marca de límite superior para las suscripciones de proxy no duraderas (MQSUBTYPE_PROXY)

Identificador: MQIAMO_SUB_NDUR_HIGHWATER

Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

SubCountNDurLowWater

Descripción: La marca de límite inferior del número de suscripciones no duraderas durante el intervalo de tiempo. Ésta es una matriz de valores indexada por el valor SUBTYPE.

0 - La marca de límite inferior para todas las suscripciones no duraderas en el sistema

1 - La marca de límite inferior para las suscripciones de aplicación no duraderas (MQSUBTYPE_API)

2 - La marca de límite inferior para las suscripciones de administración no duraderas (MQSUBTYPE_ADMIN)

3 - La marca de límite inferior para las suscripciones de proxy no duraderas (MQSUBTYPE_PROXY)

Identificador: MQIAMO_SUB_NDUR_LOWWATER

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

PutTopicCount

Descripción: El número de mensajes persistentes y no persistentes transferidos satisfactoriamente a un tema, con la excepción de los mensajes transferidos utilizando la llamada MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Nota: en este valor se incluyen los mensajes colocados mediante un alias de cola que se resuelven en un tema.

Identificador: MQIAMO_TOPIC_PUTS.

Tipo de datos: MQCFIL.

Se devuelve: Cuando esté disponible.

PutTopicFailCount

Descripción: El número de intentos de colocación de un mensaje en un tema no satisfactorios.

Identificador: MQIAMO_TOPIC_PUTS_FAILED.

Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

Put1TopicCount

Descripción: El número de mensajes persistentes y no persistentes colocados satisfactoriamente en un tema mediante llamadas MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).

Nota: en este valor se incluyen los mensajes colocados mediante un alias de cola que se resuelven en un tema.

Identificador: MQIAMO_TOPIC_PUT1S.

Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

Put1TopicFailCount

Descripción: El número de intentos de colocación de un mensaje en un tema no satisfactorios mediante llamadas MQPUT1.
Identificador: MQIAMO_TOPIC_PUT1S_FAILED.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

PutTopicBytes

Descripción: El número de bytes escritos utilizando llamadas de colocación para los mensajes persistentes y no persistentes que resuelven en una operación de tipo publish. Éste es el número de bytes que la aplicación coloca, y no el número resultante de bytes entregados a los suscriptores, consulte PublishMsgBytes para obtener información respecto a este valor. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_TOPIC_PUT_BYTES.
Tipo de datos: MQCFIL64.
Se devuelve: Cuando esté disponible.

PublishMsgCount

Descripción: El número de mensajes entregados a las suscripciones en el intervalo de tiempo. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_PUBLISH_MSG_COUNT
Tipo de datos: MQCFIL.
Se devuelve: Cuando esté disponible.

PublishMsgBytes

Descripción: El número de bytes entregados a las suscripciones en el intervalo de tiempo. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_PUBLISH_MSG_BYTES
Tipo de datos: MQCFIL64.
Se devuelve: Cuando esté disponible.

Datos de mensajes de estadísticas de cola

Utilice esta página para ver la estructura de un mensaje de estadísticas de cola.

Nombre de mensaje:	Mensaje de estadísticas de cola.
Plataformas:	Todas, excepto IBM MQ for z/OS.
Cola de sistema:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Descripción: Nombre del gestor de colas

Identificador: MQCA_Q_MGR_NAME
Tipo de datos: MQCFST
Longitud máxima: MQ_Q_MGR_NAME_LENGTH
Se devuelve: Siempre

IntervalStartDate

Descripción: La fecha del inicio del periodo de supervisión
Identificador: MQCAMO_START_DATE
Tipo de datos: MQCFST
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre

IntervalStartTime

Descripción: La hora del inicio del periodo de supervisión
Identificador: MQCAMO_START_TIME
Tipo de datos: MQCFST
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Siempre

IntervalEndDate

Descripción: La fecha del final del periodo de supervisión
Identificador: MQCAMO_END_DATE
Tipo de datos: MQCFST
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre

IntervalEndTime

Descripción: La hora del final del periodo de supervisión
Identificador: MQCAMO_END_TIME
Tipo de datos: MQCFST
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Siempre

CommandLevel

Descripción: El nivel de mandato del gestor de colas
Identificador: MQIA_COMMAND_LEVEL
Tipo de datos: MQCFIN
Se devuelve: Siempre

ObjectCount

Descripción:	El número de objetos a los que se accede en el intervalo para el que se han registrado los datos de estadísticas. El valor se establece en el número de grupos PCF QStatisticsData que el mensaje contiene.
Identificador:	MQIAMO_OBJECT_COUNT
Tipo de datos:	MQCFIN
Se devuelve:	Siempre

QStatisticsData

Descripción:	Parámetros agrupados que especifican los detalles de estadísticas de una cola
Identificador:	MQGACF_Q_STATISTICS_DATA
Tipo de datos:	MQCFGR
Parámetros del grupo:	<i>QName</i> <i>CreateDate</i> <i>CreateTime</i> <i>QType</i> <i>QDefinitionType</i> <i>QMinDepth</i> <i>QMaxDepth</i> <i>AvgTimeOnQ</i> <i>PutCount</i> <i>PutFailCount</i> <i>Put1Count</i> <i>Put1FailCount</i> <i>PutBytes</i> <i>GetCount</i> <i>GetFailCount</i> <i>GetBytes</i> <i>BrowseCount</i> <i>BrowseFailCount</i> <i>BrowseBytes</i> <i>NonQueuedMsgCount</i> <i>ExpiredMsgCount</i> <i>PurgeCount</i>
Se devuelve:	Siempre

QName

Descripción:	El nombre de la cola
Identificador:	MQCA_Q_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_Q_NAME_LENGTH
Se devuelve:	Siempre

CreateDate

Descripción:	La fecha en la que se ha creado la cola
--------------	---

Identificador: MQCA_CREATION_DATE
Tipo de datos: MQCFST
Longitud máxima: MQ_DATE_LENGTH
Se devuelve: Siempre

CreateTime

Descripción: La hora en que se ha creado la cola
Identificador: MQCA_CREATION_TIME
Tipo de datos: MQCFST
Longitud máxima: MQ_TIME_LENGTH
Se devuelve: Siempre

QType

Descripción: El tipo de la cola
Identificador: MQIA_Q_TYPE
Tipo de datos: MQCFIN
Valor: MQOT_LOCAL
Se devuelve: Siempre

QDefinitionType

Descripción: El tipo de definición de cola
Identificador: MQIA_DEFINITION_TYPE
Tipo de datos: MQCFIN
Valores: Los valores posibles son

- MQQDT_PREDEFINED
- MQQDT_PERMANENT_DYNAMIC
- MQQDT_TEMPORARY_DYNAMIC

Se devuelve: Cuando esté disponible

QMinDepth

Descripción: La profundidad de cola mínima que existe durante el período de supervisión
Identificador: MQIAMO_Q_MIN_DEPTH
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QStatisticsData*
Se devuelve: Cuando esté disponible

QMaxDepth

Descripción: La profundidad de cola máxima que existe durante el período de supervisión
Identificador: MQIAMO_Q_MAX_DEPTH
Tipo de datos: MQCFIN

Incluido en el grupo PCF: *QStatisticsData*
Se devuelve: Cuando esté disponible

AvgTimeOnQ

Descripción: El promedio de latencia, en microsegundos, de los mensajes recuperados de la cola de forma destructiva, durante el periodo de supervisión. Este parámetro es una lista de enteros indexada por valor de persistencia; consulte la [Nota de referencia 2](#).
Identificador: MQIAMO64_AVG_Q_TIME
Tipo de datos: MQCFIL64
Incluido en el grupo PCF: *QStatisticsData*
Se devuelve: Cuando esté disponible

PutCount

Descripción: El número de mensajes persistentes y no persistentes transferidos satisfactoriamente a la cola, con la excepción de las solicitudes MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia. Consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_PUTS
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QStatisticsData*
Se devuelve: Cuando esté disponible

PutFailCount

Descripción: El número de intentos de colocación de un mensaje en una cola no satisfactorios.
Identificador: MQIAMO_PUTS_FAILED
Tipo de datos: MQCFIN
Incluido en el grupo PCF: *QStatisticsData*
Se devuelve: Cuando esté disponible

Put1Count

Descripción: El número de mensajes persistentes y no persistentes colocados satisfactoriamente en la cola mediante llamadas MQPUT1. Este parámetro es una lista de enteros indexada por valor de persistencia. Consulte la [Nota de referencia 2](#).
Identificador: MQIAMO_PUT1S
Tipo de datos: MQCFIL
Incluido en el grupo PCF: *QStatisticsData*
Se devuelve: Cuando esté disponible

Put1FailCount

Descripción:	El número de intentos de colocación de un mensaje mediante llamadas MQPUT1 no satisfactorios
Identificador:	MQIAMO_PUT1S_FAILED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

PutBytes

Descripción:	El número de bytes escritos en las solicitudes de colocación a la cola
Identificador:	MQIAMO64_PUT_BYTES
Tipo de datos:	MQCFIL64
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

GetCount

Descripción:	El número de solicitudes de obtención destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia. Consulte la Nota de referencia 2 .
Identificador:	MQIAMO_GETS
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

GetFailCount

Descripción:	El número de solicitudes de obtención destructivas no satisfactorias.
Identificador:	MQIAMO_GETS_FAILED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

GetBytes

Descripción:	El número de bytes leídos en las solicitudes de colocación destructivas para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia. Consulte la Nota de referencia 2 .
Identificador:	MQIAMO64_GET_BYTES
Tipo de datos:	MQCFIL64
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

BrowseCount

Descripción:	El número de solicitudes de obtención no destructivas satisfactorias para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia. Consulte la Nota de referencia 2 .
Identificador:	MQIAMO_BROWSES
Tipo de datos:	MQCFIL
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

BrowseFailCount

Descripción:	El número de solicitudes de obtención no destructivas no satisfactorias.
Identificador:	MQIAMO_BROWSES_FAILED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

BrowseBytes

Descripción:	El número de bytes leídos en las solicitudes de obtención no destructivas para los mensajes persistentes y no persistentes. Este parámetro es una lista de enteros indexada por valor de persistencia. Consulte la Nota de referencia 2 .
Identificador:	MQIAMO64_BROWSE_BYTES
Tipo de datos:	MQCFIL64
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

NonQueuedMsgCount

Descripción:	<p>El número de mensajes que han ignorado la cola y se han transferido directamente a una aplicación que está en espera.</p> <p>El hecho de ignorar una cola sólo puede producirse en determinadas circunstancias. Este número representa cuántas veces IBM MQ ha podido ignorar la cola, y no el número de veces que una aplicación ha estado esperando.</p>
Identificador:	MQIAMO_MSGS_NOT_QUEUED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

ExpiredMsgCount

Descripción:	El número de mensajes persistentes y no persistentes que se han descartado porque habían caducado antes de que se pudieran recuperar.
--------------	---

Identificador:	MQIAMO_MSGS_EXPIRED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

PurgeCount

Descripción:	El número de mensajes depurados.
Identificador:	MQIAMO_MSGS_PURGED
Tipo de datos:	MQCFIN
Incluido en el grupo PCF:	<i>QStatisticsData</i>
Se devuelve:	Cuando esté disponible

Datos de mensajes de estadísticas de canal

Utilice esta página para ver la estructura de un mensaje de estadísticas de canal.

Nombre de mensaje:	Mensaje de estadísticas de canal.
--------------------	-----------------------------------

Plataformas:	Todas, excepto IBM MQ for z/OS.
--------------	---------------------------------

Cola de sistema:	SYSTEM.ADMIN.STATISTICS.QUEUE.
------------------	--------------------------------

QueueManager

Descripción:	Nombre del gestor de colas.
Identificador:	MQCA_Q_MGR_NAME.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH.
Se devuelve:	Siempre.

IntervalStartDate

Descripción:	La fecha del inicio del periodo de supervisión.
Identificador:	MQCAMO_START_DATE.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_DATE_LENGTH.
Se devuelve:	Siempre.

IntervalStartTime

Descripción:	La hora del inicio del periodo de supervisión.
Identificador:	MQCAMO_START_TIME.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_TIME_LENGTH.
Se devuelve:	Siempre.

IntervalEndDate

Descripción:	La fecha del final del periodo de supervisión
Identificador:	MQCAMO_END_DATE.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_DATE_LENGTH.
Se devuelve:	Siempre.

IntervalEndTime

Descripción:	La hora del final del periodo de supervisión
Identificador:	MQCAMO_END_TIME.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_TIME_LENGTH
Se devuelve:	Siempre.

CommandLevel

Descripción:	El nivel de mandato del gestor de colas.
Identificador:	MQIA_COMMAND_LEVEL.
Tipo de datos:	MQCFIN.
Se devuelve:	Siempre.

ObjectCount

Descripción:	El número de objetos de canal a los que se accede en el intervalo para el que se han registrado los datos de estadísticas. El valor se establece en el número de grupos PCF ChlStatisticsData que el mensaje contiene.
Identificador:	MQIAMO_OBJECT_COUNT
Tipo de datos:	MQCFIN.
Se devuelve:	Siempre.

ChlStatisticsData

Descripción:	Parámetros agrupados que especifican los detalles de estadísticas de un canal.
Identificador:	MQGACF_CHL_STATISTICS_DATA.
Tipo de datos:	MQCFGR.

Parámetros del grupo:

- ChannelName*
- ChannelType*
- RemoteQmgr*
- ConnectionName*
- MsgCount*
- TotalBytes*
- NetTimeMin*
- NetTimeAvg*
- NetTimeMax*
- ExitTimeMin*
- ExitTimeAvg*
- ExitTimeMax*
- FullBatchCount*
- IncplBatchCount*
- AverageBatchSize*
- PutRetryCount*

Se devuelve: Siempre.

ChannelName

Descripción: El nombre del canal.
 Identificador: MQCACH_CHANNEL_NAME.
 Tipo de datos: MQCFST.
 Longitud máxima: MQ_CHANNEL_NAME_LENGTH.
 Se devuelve: Siempre.

ChannelType

Descripción: El tipo de canal.
 Identificador: MQIACH_CHANNEL_TYPE.
 Tipo de datos: MQCFIN.
 Valores: Los valores posibles son:

- MQCHT_SENDER**
Canal emisor.
- MQCHT_SERVER**
Canal servidor.
- MQCHT_RECEIVER**
Canal receptor.
- MQCHT_REQUESTER**
Canal peticionario.
- MQCHT_CLUSRCVR**
Canal receptor de clúster.
- MQCHT_CLUSSDR**
Canal emisor de clúster.

Se devuelve: Siempre.

RemoteQmgr

Descripción: Nombre del gestor de colas remoto.

Identificador: MQCA_REMOTE_Q_MGR_NAME.
Tipo de datos: MQCFST.
Longitud máxima: MQ_Q_MGR_NAME_LENGTH
Se devuelve: Cuando esté disponible.

ConnectionName

Descripción: Nombre de conexión del gestor de colas remoto.
Identificador: MQCACH_CONNECTION_NAME.
Tipo de datos: MQCFST
Longitud máxima: MQ_CONN_NAME_LENGTH
Se devuelve: Cuando esté disponible.

MsgCount

Descripción: El número de mensajes persistentes y no persistentes enviados o recibidos.
Identificador: MQIAMO_MSGS.
Tipo de datos: MQCFIN
Se devuelve: Cuando esté disponible.

TotalBytes

Descripción: El número total de bytes enviados o recibidos para los mensajes persistentes y no persistentes.
Identificador: MQIAMO64_BYTES.
Tipo de datos: MQCFIN64.
Se devuelve: Cuando esté disponible.

NetTimeMin

Descripción: El viaje de ida y vuelta del canal registrado más corto, medido en el intervalo de registro, en microsegundos.
Identificador: MQIAMO_NET_TIME_MIN.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

NetTimeAvg

Descripción: El viaje de ida y vuelta del canal promedio registrado, medido en el intervalo de registro, en microsegundos.
Identificador: MQIAMO_NET_TIME_AVG.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

NetTimeMax

Descripción: El viaje de ida y vuelta del canal registrado más largo, medido en el intervalo de registro, en microsegundos.
Identificador: MQIAMO_NET_TIME_MAX.

Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

ExitTimeMin

Descripción: El tiempo registrado más corto, en microsegundos, empleado al ejecutar una salida de usuario en el intervalo de registro,
Identificador: MQIAMO_EXIT_TIME_MIN.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

ExitTimeAvg

Descripción: El tiempo promedio registrado, en microsegundos, empleado al ejecutar una salida de usuario en el intervalo de registro. Medido en microsegundos.
Identificador: MQIAMO_EXIT_TIME_AVG.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

ExitTimeMax

Descripción: El tiempo registrado más largo, en microsegundos, empleado al ejecutar una salida de usuario en el intervalo de registro. Medido en microsegundos.
Identificador: MQIAMO_EXIT_TIME_MAX.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

FullBatchCount

Descripción: Número de lotes procesados por el canal que se han enviado porque se ha alcanzado el valor de los atributos de canal BATCHSZ o BATCHLIM.
Identificador: MQIAMO_FULL_BATCHES.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

IncmplBatchCount

Descripción: Número de lotes procesados por el canal que se han enviado sin que se haya alcanzado el valor de los atributos de canal BATCHSZ o BATCHLIM .
Identificador: MQIAMO_INCOMPLETE_BATCHES.
Tipo de datos: MQCFIN.
Se devuelve: Cuando esté disponible.

AverageBatchSize

Descripción: El tamaño promedio de proceso por lotes, de los procesos por lotes que el canal ha procesado.
Identificador: MQIAMO_AVG_BATCH_SIZE.
Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

PutRetryCount

Descripción: El número de veces, del intervalo de tiempo, que no se ha podido transferir un mensaje, y ha entrado en un bucle de reintentos.

Identificador: MQIAMO_PUT_RETRIES.

Tipo de datos: MQCFIN.

Se devuelve: Cuando esté disponible.

Notas de referencia

Utilice esta página para ver las notas a las que hacen referencia las descripciones de la estructura de los mensajes de contabilidad y estadísticas.

Las descripciones de datos de mensaje siguientes hacen referencia a estas notas:

- [“datos de mensaje de contabilidad de MQI”](#) en la página 160
- [“Datos de mensajes de contabilidad de cola”](#) en la página 171
- [“datos de mensaje de estadísticas de MQI”](#) en la página 183
- [“Datos de mensajes de estadísticas de cola”](#) en la página 194
- [“Datos de mensajes de estadísticas de canal”](#) en la página 201

1. Este parámetro se refiere a los objetos de IBM MQ. Este parámetro es una matriz de valores (MQCFIL o MQCFIL64) que se indexa por las constantes siguientes:

Tipo de objeto	Contexto del valor
MQOT_Q (1)	Contiene el valor relativo a los objetos de cola.
MQOT_NAMELIST (2)	Contiene el valor relacionado con los objetos de lista de nombres.
MQOT_PROCESS (3)	Contiene el valor relativo a los objetos de proceso.
MQOT_Q_MGR (5)	Contiene el valor relativo a los objetos de gestor de colas.
MQOT_CHANNEL (6)	Contiene el valor relacionado con los objetos de canal.
MQOT_AUTH_INFO (7)	Contiene el valor relativo a los objetos de información de autenticación.
MQOT_TOPIC (8)	Contiene el valor relativo a los objetos de tema.

Nota: Se devuelve una matriz de 13 valores MQCFIL o MQCFIL64, pero sólo son significativos los que aparecen en la lista.

2. Este parámetro se refiere a los mensajes de IBM MQ. Este parámetro es una matriz de valores (MQCFIL o MQCFIL64) que se indexa por las constantes siguientes:

Constante	Valor
1	Contiene el valor para los mensajes no persistentes.

<i>Tabla 25. Matriz indexada por valor de persistencia (continuación)</i>	
Constante	Valor
2	Contiene el valor para los mensajes persistentes.

Nota: El índice de cada una de estas matrices empieza en cero, por lo que un índice de 1 hace referencia a la segunda fila de la matriz. Los elementos de estas matrices que no aparecen en estas tablas no contienen ninguna información de contabilidad ni estadísticas.

Rastreo de actividad de la aplicación

El rastreo de actividad de la aplicación genera información detallada sobre el comportamiento de las aplicaciones conectadas a un gestor de colas. Rastrea el comportamiento de una aplicación y proporciona una vista detallada de los parámetros utilizados por una aplicación a medida que interactúa con los recursos de IBM MQ. También muestra la secuencia de llamadas MQI emitidas por una aplicación.


Utilice el rastreo de actividad de la aplicación cuando necesite más información de la que proporciona la supervisión de sucesos, la supervisión de mensajes, los mensajes de contabilidad y estadísticas y la supervisión en tiempo real.

Nota: El rastreo de actividad se genera en la conexión IBM MQ para cada aplicación; por lo tanto, si los mensajes de contabilidad están habilitados, las operaciones de rastreo de actividad cuentan para la información de contabilidad MQI para cada aplicación.

IBM MQ soporta dos métodos para recopilar datos de rastreo de actividad de aplicación.

- Recopilación central de información de rastreo de actividad de aplicación, donde el rastreo de actividad de aplicación se recopila y lee escribiendo mensajes PCF de rastreo de actividad en la cola de sistema SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.
- Suscripción a datos de rastreo de la actividad, escritos en temas especiales del sistema IBM MQ.

Tenga en cuenta que IBM MQ for z/OS no admite este rastreo de actividad.

 Puesto que ahora puede especificar nombres de aplicación en la mayoría de los lenguajes de programación que IBM MQ admite, consulte [Especificación del nombre de aplicación en lenguajes de programación admitidos](#) para obtener más información.

Configuración de la recopilación central de información de rastreo de actividad de aplicaciones

Un mensaje de rastreo de actividad de la aplicación es un mensaje PCF. El rastreo de actividad se configura utilizando un archivo de configuración. Para configurar la recopilación central de información de rastreo de actividad de aplicaciones, se define el atributo de gestor de colas ACTVTRC. Puede alterar temporalmente este valor a nivel de conexión utilizando las opciones MQCONNX, o a nivel de stanza de aplicación utilizando el archivo de configuración de rastreo de actividad.

Acerca de esta tarea

Los mensajes de rastreo de actividad están formados por una estructura MQMD: una estructura de cabecera PCF (MQCFH), seguida de varios parámetros PCF. Los parámetros PCF van seguidos de una secuencia de grupos PCF ApplicationTraceData. Estos grupos PCF recopilan información sobre las operaciones MQI que una aplicación realiza mientras está conectada a un gestor de colas. Puede configurar el rastreo de actividad utilizando un archivo de configuración denominado `mqat.ini`.

Para controlar si se recopila o no información de rastreo de actividad de la aplicación, debe configurar uno o más de los siguientes valores:

1. El atributo de gestor de colas ACTVTRC.
2. Los valores de ACTVCONO (en la estructura MQCNO pasada en MQCONNX).

3. La stanza coincidente para la aplicación en el archivo de configuración de rastreo de actividad `mqt.ini`.

La secuencia anterior es significativa. El atributo `ACTVTRC` se altera temporalmente mediante los valores de `ACTVCONO`, que se alteran temporalmente mediante los valores del archivo `mqt.ini`.

Se graban entradas de rastreo una vez que se ha completado cada operación, a menos que se indique lo contrario. Estas entradas se graban primero en la cola de sistema `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`, y luego se graban en mensajes de rastreo de actividad de la aplicación cuando la aplicación se desconecta del gestor de colas. Para las aplicaciones de larga ejecución, se graban mensajes intermedios si se produce cualquiera de los siguientes sucesos:

- El tiempo de vida de la conexión alcanza un valor de tiempo de espera definido.
- El número de operaciones alcanza un número especificado.
- La cantidad de datos recopilados en memoria alcanza la longitud máxima de mensaje permitida para la cola.

El valor de tiempo de espera se establece mediante el parámetro **ActivityInterval**. El número de operaciones se establece mediante el parámetro **ActivityCount**. Ambos parámetros se especifican en el archivo de configuración de rastreo de actividad `mqt.ini`.

La habilitación del rastreo de actividad de la aplicación puede afectar al rendimiento. La sobrecarga se puede reducir ajustando los valores de **ActivityCount** y **ActivityInterval**. Consulte [“Ajustar el impacto en el rendimiento del rastreo de actividad de la aplicación”](#) en la página 216

La forma más fácil de ver el contenido de los mensajes de rastreo de actividad de la aplicación es utilizar el [“Programa de ejemplo amqsact”](#) en la página 216.

Procedimiento

1. [“Establecer ACTVTRC para controlar la recopilación de información de rastreo de actividad”](#) en la página 208.
2. [“Establecer opciones MQCONNX para controlar la recopilación de información de rastreo de actividad”](#) en la página 209.
3. [“Configuración del comportamiento de rastreo de actividad utilizando mqt.ini”](#) en la página 210.
4. [“Ajustar el impacto en el rendimiento del rastreo de actividad de la aplicación”](#) en la página 216.

Establecer ACTVTRC para controlar la recopilación de información de rastreo de actividad

Utilice el atributo de gestor de colas `ACTVTRC` para controlar la recopilación de información de rastreo de actividad de aplicaciones MQI.

Acerca de esta tarea

Los mensajes de rastreo de actividad de la aplicación sólo se generan para las conexiones que empiecen después de habilitarse el rastreo de actividad de la aplicación. El parámetro **ACTVTRC** puede tener los valores siguientes:

ON

La recopilación de rastreo de actividad del API está habilitada

OFF

La recopilación de rastreo de actividad del API está inhabilitada

Nota: El valor de **ACTVTRC** se puede alterar temporalmente con el parámetro **ACTVCONO** del gestor de colas. Si establece el parámetro **ACTVCONO** en `ENABLED`, el valor de **ACTVTRC** se puede alterar temporalmente para una conexión determinada utilizando el campo **Options** de la estructura `MQCNO`. Consulte [“Establecer opciones MQCONNX para controlar la recopilación de información de rastreo de actividad”](#) en la página 209.

Ejemplo

Para cambiar el valor del parámetro **ACTVTRC**, utilice el mandato MQSC ALTER QMGR. Por ejemplo, para habilitar la recopilación de información de rastreo de actividad de aplicaciones MQI, utilice el siguiente mandato MQSC:

```
ALTER QMGR ACTVTRC(ON)
```

Qué hacer a continuación

La forma más fácil de ver el contenido de los mensajes de rastreo de actividad de la aplicación es utilizar el [“Programa de ejemplo amqsact”](#) en la página 216.

La habilitación del rastreo de actividad de la aplicación puede afectar al rendimiento. La sobrecarga se puede reducir ajustando los valores de **ActivityCount** y **ActivityInterval**. Consulte [“Ajustar el impacto en el rendimiento del rastreo de actividad de la aplicación”](#) en la página 216

Establecer opciones MQCONNX para controlar la recopilación de información de rastreo de actividad

Si el atributo de gestor de colas **ACTVCONO** está establecido en ENABLED, puede utilizar el parámetro **ConnectOpts** de la llamada MQCONNX para habilitar o inhabilitar informes de actividad de la aplicación para cada conexión. Estas opciones alteran temporalmente el comportamiento de rastreo de actividad definido mediante el atributo de gestor de colas **ACTVTRC**, y se pueden alterar temporalmente con valores del archivo de configuración de rastreo de actividad `mqt.ini`.

Procedimiento

1. Establezca el atributo de gestor de colas **ACTVCONO** en ENABLED.

Nota: Si una aplicación intenta modificar el comportamiento de contabilidad de una aplicación utilizando el parámetro **ConnectOpts**, y el atributo QMGR **ACTVCONO** está establecido en DISABLED, no se devuelve ningún error a la aplicación, y la recopilación de rastreo de actividad se define mediante los atributos de gestor de colas o el archivo de configuración de rastreo de actividad `mqt.ini`.

2. Establezca el parámetro **ConnectOpts** de la llamada MQCONNX en MQCNO_ACTIVITY_TRACE_ENABLED.

El parámetro **ConnectOpts** de la llamada MQCONNX puede tener los siguientes valores:

MQCNO_ACTIVITY_TRACE_DISABLED

El rastreo de actividad está inhabilitado en la conexión.

MQCNO_ACTIVITY_TRACE_ENABLED

El rastreo de actividad está habilitado en la conexión.

Nota: Si una aplicación selecciona ambos valores, MQCNO_ACTIVITY_TRACE_ENABLED y MQCNO_ACTIVITY_TRACE_DISABLED, para MQCONNX, la llamada falla con un código de razón MQRC_OPTIONS_ERROR.

3. Compruebe que estos valores de rastreo de actividad no se alteren temporalmente mediante valores en el archivo de configuración de rastreo de actividad `mqt.ini`.

Consulte [“Configuración del comportamiento de rastreo de actividad utilizando mqt.ini”](#) en la página 210.

Qué hacer a continuación

La forma más fácil de ver el contenido de los mensajes de rastreo de actividad de la aplicación es utilizar el [“Programa de ejemplo amqsact”](#) en la página 216.

La habilitación del rastreo de actividad de la aplicación puede afectar al rendimiento. La sobrecarga se puede reducir ajustando los valores de **ActivityCount** y **ActivityInterval**. Consulte [“Ajustar el impacto en el rendimiento del rastreo de actividad de la aplicación”](#) en la página 216

Configuración del comportamiento de rastreo de actividad utilizando mqat.ini

El comportamiento de rastreo de actividad se configura utilizando un archivo de configuración denominado mqat.ini. Este archivo se utiliza para definir el nivel y la frecuencia de notificación de datos de rastreo de actividad. El archivo también proporciona una forma de definir reglas para habilitar e inhabilitar el rastreo de actividad basándose en el nombre de una aplicación.

Acerca de esta tarea

Linux **AIX** En sistemas AIX and Linux, mqat.ini se encuentra en el directorio de datos del gestor de colas, que es la misma ubicación que el archivo qm.ini.

Windows En sistemas Windows, mqat.ini se encuentra en el directorio de datos del gestor de colas de C:\Program Files\IBM\WebSphere MQ\mqmgs\queue_manager_name. Los usuarios que ejecutan aplicaciones que hay que rastrear necesitan permiso para leer este archivo.

Nota: Los gestores de colas migrados de IBM WebSphere MQ 7.1 o anteriores tendrán el archivo mqat.ini que falta. En estos casos, es necesario crear manualmente el archivo mqat.ini y hay que establecer 660 permisos en el archivo.

Cuando se modifique el archivo mqat.ini, las conexiones IBM MQ recién creadas se procesarán de acuerdo con la versión modificada. Las conexiones existentes seguirán utilizando la versión anterior, a menos que se modifiquen los parámetros del gestor de colas, por ejemplo después de un mandato [ALTER QMGR](#).

Este archivo sigue la misma clave de stanza y el mismo formato de parámetro-valor que los archivos mqs.ini y qm.ini.

El archivo consta de una sola stanza, **AllActivityTrace**, para configurar el nivel y la frecuencia de la notificación de datos de rastreo de actividad de forma predeterminada para todo rastreo de actividad.

El archivo también puede contener varias stanzas **ApplicationTrace**. Cada una de estas stanzas define una regla para el comportamiento de rastreo para una o más conexiones, que se basa en la coincidencia del nombre de aplicación de las conexiones con la regla.

Stanza AllActivityTrace

Una stanza **AllActivityTrace** define valores para el rastreo de actividad que se aplica a todas las conexiones de IBM MQ, a menos que se alteren temporalmente.

Los valores individuales de la stanza **AllActivityTrace** se pueden alterar temporalmente con información más específica en una stanza **ApplicationTrace**.

Si se especifica más de una stanza **AllActivityTrace**, se utilizan los valores de la última stanza. Los parámetros faltantes en la stanza **AllActivityTrace** elegida adoptan los valores predeterminados. Los parámetros y valores de las stanzas **AllActivityTrace** anteriores se ignoran.

Los siguientes parámetros se pueden especificar en la stanza **AllActivityTrace**:

Tabla 26. Pares de parámetro/valor que se pueden utilizar en el archivo de configuración de rastreo de actividad

Nombre	Valores (valor predeterminado en tipo negrita)	Descripción
ActivityInterval	0-99999999 (1)	Intervalo de tiempo aproximado, en segundos, entre mensajes de rastreo. Toda la actividad realizada por una conexión en ese intervalo se grabará en un único mensaje. Si este valor es 0, el mensaje de rastreo se graba cuando la conexión se desconecta (o cuando se alcanza el valor especificado en el parámetro ActivityCount).
ActivityCount	0-99999999 (100)	Número de operaciones MQI o XA entre mensajes de rastreo. Si este valor es 0, el mensaje de rastreo se graba cuando la conexión se desconecta (o cuando ha transcurrido el valor especificado en el parámetro ActivityInterval).
TraceLevel	LOW / MEDIUM / HIGH	El nivel de detalle de parámetro rastreado para cada operación. La descripción de las operaciones individuales detalla qué parámetros se incluyen para cada nivel de rastreo.
TraceMessageData	0 - 104857600 (máximo 100 MB)	Cantidad de datos de mensaje rastreados en bytes para operaciones MQGET, MQPUT, MQPUT1 y de devolución de llamada
StopOnGetTraceMsg	ON / OFF	No es aconsejable utilizar el rastreo de actividad para rastrear aplicaciones que también están procesando mensajes de rastreo de actividad debido a la posibilidad de que se produzca un bucle.
SubscriptionDelivery	POR lotes /IMMEDIATE	Determina si los parámetros ActivityInterval y ActivityCount se utilizarán cuando estén presentes una o más suscripciones de rastreo de actividad. Establecer este parámetro a IMMEDIATE hace que los valores ActivityInterval y ActivityCount se sustituyan con valores efectivos de 1 cuando los datos de rastreo tienen una suscripción coincidente. Cada registro de rastreo de actividad no se procesa por lotes con otros registros de la misma conexión y en lugar de ello se entrega a la suscripción inmediatamente sin retardo.

Stanza ApplicationTrace

Una stanza ApplicationTrace contiene una regla que define qué conexiones de IBM MQ se rastrearán o no basándose en el nombre de la aplicación. Opcionalmente, el comportamiento predeterminado definido en los valores, que alteran temporalmente los valores de nivel y frecuencia de rastreo globales.

Esta stanza puede incluir parámetros ApplName, ApplFunction y ApplClass, que se utilizan de acuerdo con las reglas de coincidencia definidas en las Reglas de coincidencia de conexión para determinar si la stanza se aplica a una conexión específica o no.

La stanza debe incluir el parámetro Trace para determinar si esta regla activa o desactiva el rastreo de actividad para las conexiones coincidentes.

Se puede utilizar una regla de desactivación ('off') para inhabilitar explícitamente el rastreo para nombres de aplicación más específicos y para alterar temporalmente el valor ACTVTRC del gestor de colas o las opciones de conexión de rastreo de actividad.

Los siguientes parámetros se pueden especificar en la stanza **ApplicationTrace**:




<i>Tabla 27. Pares de parámetro/valor que se pueden utilizar en el archivo de configuración de rastreo de aplicaciones</i>		
Nombre	Valores (valor predeterminado en tipo negrita)	Descripción
Rastreo	ON / OFF (parámetro necesario; ningún valor predeterminado)	Conmutador de rastreo de actividad. Este conmutador se puede utilizar en la stanza para determinar si el rastreo de la actividad está activo para el ámbito de la stanza de aplicación actual. Tenga en cuenta que este valor altera temporalmente los valores de ACTVTRC y ACTVCONO para el gestor de colas.
ApplName	Serie de caracteres (parámetro necesario; ningún valor predeterminado)	<p>Este valor se utiliza para determinar a qué aplicaciones se aplica la stanza ApplicationTrace. Se compara con el valor ApplName de la estructura de contexto de salida de la API (que es equivalente al MQMD.PutApplName). El contenido del valor ApplName varía en función del entorno de la aplicación.</p> <p>Para las plataformas que no son z/OS, sólo la parte de MQAXC.ApplName correspondiente al nombre de archivo se compara con el valor de la stanza. Los caracteres a la izquierda del separador de vía de acceso situado en el extremo derecho se ignoran cuando se efectúa la comparación.</p> <p> Para las aplicaciones z/OS, toda la MQAXC.ApplName se compara con el valor de la stanza.</p> <p>Se puede utilizar un único carácter comodín (*) al final del valor de ApplName para que coincida con cualquier número de caracteres después de ese punto. Si el valor ApplName está establecido en un único carácter comodín (*), el valor ApplName coincide con todas las aplicaciones.</p>

Tabla 27. Pares de parámetro/valor que se pueden utilizar en el archivo de configuración de rastreo de aplicaciones (continuación)

Nombre	Valores (valor predeterminado en tipo negrita)	Descripción
  ApplFunction	Serie de caracteres (valor predeterminado *)	Este valor se utiliza para determinar a qué programas de aplicación se aplica la stanza ApplicationTrace y el valor App1Name . La stanza es opcional, pero sólo es válida para los gestores de colas de IBM i. Se puede utilizar un único carácter comodín (*) al final del valor de App1Name para que coincida con cualquier número de caracteres. Por ejemplo, una stanza ApplicationTrace que especifique App1Name=* y App1Function=AMQSPUTO se aplica a todas las invocaciones del programa AMQSPUTO de cualquier trabajo.
App1Class	USER / MCA / ALL	La clase de aplicación. Consulte la siguiente tabla para ver una explicación de cómo los valores de AppType se corresponden con las conexiones de IBM MQ.

La siguiente tabla muestra cómo los valores de *AppClass* se corresponden con los campos **APICallerType** y **APIEnvironment** en la estructura de contexto de salida de API de la conexión.

Tabla 28. Los valores de *Appclass* y cómo se corresponden a los campos *APICallerType* y *APIEnvironment*

APPLCLASS	Tipo de interlocutor de la API:	Entorno de API:	Descripción
USER	MQXACT_EXTERNAL	MQXE_OTHER	Sólo se rastrean las aplicaciones de usuario
MCA	(Cualquier valor)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	Clientes y canales (amqrmppa)
TODOS	(Cualquier valor)	(Cualquier valor)	Se rastrean todas las conexiones



Atención: Debe utilizar un **APPLCLASS** de *MCA* para aplicaciones de usuario de cliente, porque una clase de *USER* no coincide con éstas.

Por ejemplo, para rastrear la aplicación de ejemplo **amqspu~~t~~c**, puede utilizar el código siguiente:

```

ApplicationTrace:
App1Class=MCA                                     # Application type
                                                # Values: (USER | MCA | INTERNAL | ALL)
                                                # Default: USER
App1Name=amqsputc      # Application name (may be wildcard)
                                                # (matched to app name without path)
                                                # Default: *
Trace=ON                                           # Activity trace switch for application
                                                # Values: ( ON | OFF )
                                                # Default: OFF
ActivityInterval=30                               # Time interval between trace messages
                                                # Values: 0-99999999 (0=off)
                                                # Default: 0
ActivityCount=1                                  # Number of operations between trace msgs
                                                # Values: 0-99999999 (0=off)
                                                # Default: 0
    
```

```

TraceLevel=MEDIUM           # Amount of data traced for each operation
                             # Values: LOW | MEDIUM | HIGH
                             # Default: MEDIUM
TraceMessageData=1000      # Amount of message data traced
                             # Values: 0-1000000000
                             # Default: 0

```

El archivo `mqat.ini` predeterminado, que se genera cuando se crea un gestor de colas, contiene una sola regla para inhabilitar explícitamente el rastreo de actividad para el ejemplo de rastreo de actividad suministrado, **amqsact**.

Reglas de coincidencia de conexión

El gestor de colas aplica las siguientes reglas para determinar qué valores de stanza se han de utilizar para una conexión.

1. Un valor especificado en la stanza **AllActivityTrace** se utiliza para la conexión, a menos que el valor también aparezca en una stanza **ApplicationTrace** y la stanza cumpla los criterios de coincidencia para la conexión descritos en los puntos 2, 3 y 4.
2. La **App1Class** se compara con el tipo de la conexión de IBM MQ. Si la **App1Class** no coincide con el tipo de conexión, la stanza se ignora para esta conexión.
3. El valor de *App1Name* en la stanza se compara con la parte correspondiente al nombre de archivo del campo **App1Name** de la estructura de contexto de salida de API (MQAXC) para la conexión.

La parte del nombre de archivo se deriva de los caracteres a la derecha de los caracteres del separador de vía de acceso finales (/ o \). Si la stanza **App1Name** incluye un comodín (*), sólo los caracteres a la izquierda del comodín se comparan con el número equivalente de caracteres del campo **App1Name** de la conexión.

Por ejemplo, si se especifica el valor de stanza "FRE*", en la comparación sólo se utilizarán los tres primeros caracteres, por lo que "path/FREEDOM" y "path\FREDDY" coincidirán, pero no lo hará "path/FRIEND". Si el valor de *App1Name* de la stanza no coincide con el **App1Name** de la conexión, la stanza se ignora para esta conexión.

4. Si más de una stanza coincide con los valores de **App1Name** y **App1Class** de la conexión, se utiliza la stanza con el valor de **App1Name** más específico.

El valor de **App1Name** más específico se define como el que utiliza el máximo número de caracteres para que coincidan con el valor de **App1Name** de la conexión.

Por ejemplo, si el archivo `ini` contiene una stanza con **App1Name**="FRE*" y otra stanza con **App1Name**="FREE*", la stanza con **App1Name**="FREE*" se elige como la mejor coincidencia para una conexión con **App1Name**="path/FREEDOM", ya que coinciden con cuatro caracteres (mientras que **App1Name**="FRE*" sólo coincide con tres).

5. Si después de aplicar las reglas de los puntos 2, 3 y 4, hay más de una stanza que coincide con el **App1Name** y **App1Class** de la conexión, se utilizarán los valores de la última coincidencia y todas las demás stanzas se ignorarán.

Alteración temporal de los valores predeterminados para cada regla

Opcionalmente, los valores de nivel y frecuencia de rastreo globales en la stanza **AllActivityTrace** se pueden alterar temporalmente para aquellas conexiones que coincidan con una stanza **ApplicationTrace**.

Los siguientes parámetros se pueden establecer en una stanza **ApplicationTrace**. Si no se establecen, el valor se hereda de los valores de la stanza **AllActivityTrace**:

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**
- **StopOnTraceMsg**

Sintaxis de `mqat.ini`

Las reglas de sintaxis para el formato del archivo `mqat.ini` son:

- El texto que empieza con un guión o un punto y coma se considera que es un comentario que se extiende hasta el final de la línea.
- La primera línea significativa (no es un comentario) debe ser una clave de stanza.
- Una clave de stanza consiste en el nombre de la stanza seguido de dos puntos.
- Una par parámetro-valor consta del nombre de un parámetro seguido por un signo igual y luego el valor.
- Sólo puede aparecer en una línea un par parámetro-valor. (Un parámetro-valor no debe pasar a otra línea).
- Se ignoran los espacios en blanco iniciales y finales. No hay ningún límite en la cantidad de espacios en blanco entre los nombres de stanza, nombres y valores de parámetro, o pares parámetro/valor. Los saltos de línea son significativos y no se ignoran
- La longitud máxima para cada línea es de 2048 caracteres
- Las claves de stanza, nombres de parámetro y valores de parámetro de constantes no son sensibles a las mayúsculas y minúsculas, pero los valores de parámetros variables (*ApplName* y *DebugPath*) son sensibles a las mayúsculas y minúsculas.

Ejemplo de archivo de rastreo de actividad de la aplicación

El siguiente ejemplo muestra cómo se especifican los datos de configuración en el archivo `ini` de rastreo de actividad.

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
ApplName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
ApplName=amqsact*
Trace=OFF
```

La stanza **AllActivityTrace** anterior define cómo actuará el rastreo de actividad de forma predeterminada cuando se habilite mediante reglas de `ApplicationTrace` o mediante el atributo de gestor de colas `ACTVTRC`, o cuando se habilite programáticamente mediante una aplicación.

La primera stanza **ApplicationTrace** define una regla que dará como resultado el rastreo de cualquier actividad MQI realizada por una aplicación cuyo nombre empiece por "amqs". El rastreo generado para estas aplicaciones tendrá un alto nivel de detalle e incluirá hasta 1000 bytes de datos de mensaje. Los parámetros de intervalo y recuento de actividad (`ActivityInterval` y `ActivityCount`) se heredarán.

La segunda stanza **ApplicationTrace** define una regla que desactiva el rastreo para las aplicaciones con nombres que empiecen por "amqsact" (el ejemplo de rastreo de actividad). Esta regla alterará temporalmente la regla de activación ('on') anterior para la aplicación `amqsact`, dando como resultado que no haya ningún rastreo para esa aplicación.

Un ejemplo también se envía como un ejemplo denominado `mqat.ini` en el directorio de ejemplos de C (el mismo directorio que el archivo `amqsact.c`). Este archivo se puede copiar en el directorio de datos de gestor de colas, para los gestores de colas que se hayan migrado desde un release anterior de IBM MQ.

Qué hacer a continuación

La habilitación del rastreo de actividad de la aplicación puede afectar al rendimiento. La sobrecarga se puede reducir ajustando los valores de **ActivityCount** y **ActivityInterval**. Consulte ["Ajustar el impacto en el rendimiento del rastreo de actividad de la aplicación"](#) en la página 216

Ajustar el impacto en el rendimiento del rastreo de actividad de la aplicación

La habilitación del rastreo de actividad de la aplicación puede tener un impacto negativo en el rendimiento. Este impacto se puede reducir rastreando solamente las aplicaciones que necesite, aumentando el número de aplicaciones que drenan la cola y ajustando los valores de **ActivityInterval**, **ActivityCount** y **TraceLevel** en el archivo `mqt.ini`.

Acerca de esta tarea

Habilitar el rastreo de actividad de la aplicación de forma selectiva para una aplicación o para todas las aplicaciones de gestor de colas puede dar como resultado actividad de mensajería adicional y que el gestor de colas necesite espacio de almacenamiento adicional. En entornos en los que el rendimiento de la mensajería es fundamental, por ejemplo, en aplicaciones con una gran carga de trabajo o donde un acuerdo de nivel de servicio (SLA) requiere un tiempo de respuesta mínimo del proveedor de mensajería, podría no ser conveniente recopilar el rastreo de actividad de la aplicación o podría ser necesario ajustar el detalle o la frecuencia de los mensajes de actividad de rastreo que se generan. Los valores preestablecidos de **ActivityInterval**, **ActivityCount** y **TraceLevel** en el archivo `mqt.ini` proporcionan un equilibrio predeterminado de detalle y rendimiento. No obstante, puede ajustar estos valores para cumplir los requisitos funcionales y de rendimiento precisos del sistema.

Procedimiento

- Rastree únicamente las aplicaciones que necesite.

Puede hacer esto creando una stanza `ApplicationTrace` específica de la aplicación en `mqt.ini`, o modificando la aplicación para especificar `MQCNO_ACTIVITY_TRACE_ENABLED` en el campo de opciones de la estructura **MQCNO** en una llamada `MQCONN`. Consulte [“Configuración del comportamiento de rastreo de actividad utilizando mqt.ini”](#) en la página 210 y [“Establecer opciones MQCONN para controlar la recopilación de información de rastreo de actividad”](#) en la página 209.

- Antes de iniciar el rastreo, compruebe que haya al menos una aplicación en ejecución y que esté preparada para recuperar los datos de mensaje de rastreo de actividad de la cola `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`.
- Mantenga la profundidad de cola lo más baja posible aumentando el número de aplicaciones que drenan la cola.
- Establezca el valor de **TraceLevel** en el archivo `mqt.ini` para recopilar la cantidad mínima de datos necesarios.
`TraceLevel=LOW` tiene el menor impacto en el rendimiento de la mensajería. Consulte [“Configuración del comportamiento de rastreo de actividad utilizando mqt.ini”](#) en la página 210.
- Ajuste los valores de **ActivityCount** y **ActivityInterval** en el archivo `mqt.ini` para ajustar la frecuencia con la que se generan mensajes de rastreo de actividad.

Si va a rastrear varias aplicaciones, puede que los mensajes de rastreo de actividad se generen a un ritmo más rápido del que pueden eliminarse de la cola `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`. Sin embargo, cuando se reduce la frecuencia con la que se generan los mensajes de rastreo de actividad, también se aumenta el espacio de almacenamiento que requiere el gestor de colas y el tamaño de los mensajes cuando se graban en la cola.




Qué hacer a continuación

La forma más fácil de ver el contenido de los mensajes de rastreo de actividad de la aplicación es utilizar el [“Programa de ejemplo `amqsact`”](#) en la página 216.

Programa de ejemplo `amqsact`

amqsact da formato a los mensajes de Rastreo de actividad de la aplicación de forma automática y se proporciona con IBM MQ.

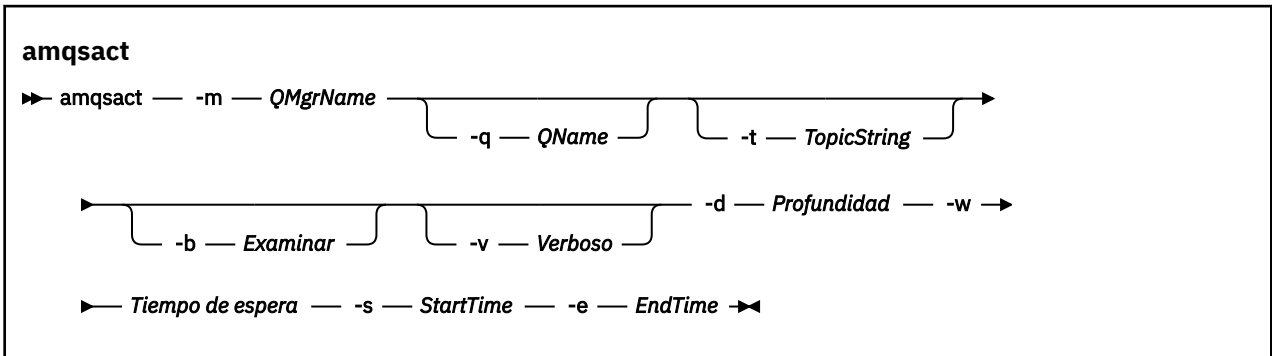
El programa compilado se encuentra en el directorio de ejemplos:

-   En AIX and Linux MQ_INSTALLATION_PATH/samp/bin
-  En Windows MQ_INSTALLATION_PATH\tools\c\Samples\Bin

Modalidad de visualización

De forma predeterminada, **amqsact** en modalidad de visualización procesa los mensajes en SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Puede alterar temporalmente este comportamiento especificando un nombre de cola o una serie de tema.

También puede controlar el período de rastreo visualizado y especificar si los mensajes de rastreo de actividad se eliminan o se conservan después de su visualización.



Parámetros necesarios para la modalidad de visualización

-m NombreGestColas

Nombre del gestor de colas.

-d Depth

Número de registros a visualizar.

-w TiempoEspera

Tiempo de espera, en segundos. Si no aparece ningún mensaje de rastreo en el período especificado, el programa **amqsact** finaliza.

-s StartTime

Hora de inicio del registro a procesar.

-e EndTime

Hora de finalización del registro a procesar.

Parámetros opcionales para la modalidad de visualización

-q QName

Especifique una cola determinada para alterar temporalmente el nombre de cola predeterminado.

-t SerieTema

Suscribirse a un tema de suceso

-b

Examinar sólo registros

-v

Salida detallada

Salida de ejemplo para la modalidad de visualización

Utilizar **amqsact** en el gestor de colas *TESTQM*, con salida detallada, en una llamada de API MQCONN:

```
amqsact -m TESTQM -v
```

El mandato anterior produce la siguiente salida de ejemplo:

Parámetros opcionales para la modalidad dinámica

-a *Nombre de aplicación*

Especifique el nombre de una aplicación para la que recopilar mensajes.

-c *Nombre de canal*

Especifique un canal para el que recopilar mensajes.

-i *ID de conexión*

Especifique una conexión para la que recopilar mensajes.

-v

Salida detallada

Salida de ejemplo para la modalidad dinámica

El siguiente mandato genera y muestra mensajes de rastreo de actividad para las conexiones realizadas por las aplicaciones cuyo nombre empieza con el texto "amqs". Después de 30 segundos de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

El siguiente mandato genera y muestra mensajes de rastreo de actividad para cualquier actividad en el canal QMGR1.TO.QMGR2. Después de 10 segundos de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

El siguiente mandato genera y muestra mensajes de rastreo de actividad detallados para cualquier actividad en la conexión de IBM MQ existente que tiene un CONN de "6B576B5420000701" y un EXTCONN de "414D5143514D47523120202020202020". Después de un minuto de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

Suscripción a la información de rastreo de actividad de una aplicación

Puede suscribirse dinámicamente a la información de rastreo de actividad de aplicación como alternativa a la recopilación de la información a través de la configuración de nivel de gestor de colas.

Acerca de esta tarea

El rastreo de actividad de aplicación rastrea el comportamiento de una aplicación y proporciona una vista detallada de los parámetros utilizados por una aplicación a medida que interactúa con los recursos de IBM MQ. También muestra la secuencia de llamadas MQI emitidas por una aplicación.

Además de escribir datos de rastreo en la cola del sistema, el producto proporciona la posibilidad de suscribirse dinámicamente a datos de rastreo de actividad, escritos en temas especiales del sistema IBM MQ, en lugar de recopilar la información a través de la configuración de nivel de gestor de colas.

Tenga en cuenta que desde IBM MQ 9.0 el producto no utiliza salidas para este fin. Si ha utilizado anteriormente salidas para rastrear la actividad de la aplicación, debe conmutar a utilizar el método de sustitución para recopilar el rastreo de la actividad de la aplicación.

La creación de una suscripción permite el rastreo de actividad. No tiene que establecer atributos de gestor de colas o aplicación como para la recopilación central de datos de rastreo. Sin embargo, cualquier bloqueo explícito del rastreo de actividad mediante la inhabilitación del rastreo en los niveles de gestor de colas o aplicación también impide que el rastreo de actividad se entregue a las suscripciones coincidentes.

Procedimiento

- [“Suscripciones al rastreo de actividad de aplicación” en la página 220](#)
- [“Creación de suscripciones al rastreo de actividad de aplicación” en la página 220](#)
- [“Utilización de amqsact para ver mensajes de rastreo” en la página 222](#)
- [“Configuración de los niveles de rastreo utilizando mqat.ini” en la página 224](#)

Suscripciones al rastreo de actividad de aplicación

Puede suscribirse a un tema de sistema IBM MQ para recopilar información de rastreo de actividad de aplicación.

Se suscribe a una serie de temas de sistema IBM MQ especial que representa la actividad a rastrear. La suscripción genera automáticamente mensajes de datos de rastreo de actividad y los publica en la cola de destino de suscripción. Si suprime la suscripción, la generación de datos de rastreo de actividad se detiene para esa suscripción.

Una suscripción puede rastrear la actividad en uno de los siguientes recursos:

- Una aplicación especificada
- Un canal de IBM MQ especificado
- Una conexión de IBM MQ existente

Puede crear varias suscripciones, con series de tema diferentes o iguales. Cuando cree varias suscripciones con las mismas series de tema de rastreo de actividad de sistema, cada suscripción recibe una copia de los datos de rastreo de actividad y esto puede tener implicaciones adversas en el rendimiento.

La habilitación de cualquier nivel de rastreo de actividad puede tener efectos adversos en el rendimiento. A cuantas más suscripciones o cuantos más recursos se haya suscrito, mayor es la potencial sobrecarga en el rendimiento. Para minimizar la sobrecarga de recopilar el rastreo de actividad, los datos se graban en mensajes y se entregan a las suscripciones de forma asíncrona desde la propia actividad de aplicación. A menudo, se graban varias operaciones en un único mensaje de datos de rastreo de actividad. La operación asíncrona puede introducir un retardo entre la operación de aplicación y la recepción de los datos de rastreo que registran la operación.

Creación de suscripciones al rastreo de actividad de aplicación

Puede crear suscripciones a temas específicos para recopilar datos de rastreo de actividad de aplicación.

Cuando se crea una suscripción para series de tema de sistema específicas, se publican automáticamente mensajes de datos PCF de rastreo de actividad adecuados para esa suscripción. Para obtener información detallada sobre la suscripción a temas, consulte [Mensajería de publicación/suscripción](#).

Las series de tema tienen el formato:

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifier
```

Donde:

- *nombreGC* especifica el gestor de colas al que está conectada la aplicación rastreada. *qmgr_name* es el nombre del gestor de colas con todos los caracteres en blanco finales eliminados y los caracteres de barra inclinada (/) sustituidos por un ampersand (&).
- *tipo_recurso* especifica el tipo de datos de recurso que se están recopilando y es una de las series siguientes:
 - *App1Name* para especificar una aplicación. La solicitud se suscribe a todas las conexiones de IBM MQ que tienen un nombre de aplicación que coincide con el especificado por el *identificador_recurso*.
 - *ChannelName* para especificar un canal de IBM MQ.
 - *ConnectionId* para especificar una conexión de IBM MQ.
- *identificador_recurso* identifica el recurso real. El formato depende del tipo de recurso:

El ejemplo siguiente muestra una serie de tema que crea una suscripción a datos de rastreo para las aplicaciones con nombres que empiezan por "amqs" (tenga en cuenta que para utilizar el comodín "*", la suscripción debe crearse utilizando el modelo de comodín de carácter):

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqs*
```

Conceptos relacionados

“Temas del sistema para la supervisión y el rastreo de actividades” en la página 295

Los temas del sistema en los árboles de temas del gestor de colas se utilizan para la supervisión de recursos (algunos de los cuales son similares al contenido de los mensajes de estadísticas) y como una forma de consumir el rastreo de actividad de la aplicación.

Utilización de **amqsact** para ver mensajes de rastreo

Puede utilizar el programa **amqsact** para generar y ver mensajes de rastreo.

El programa **amqsact** es un ejemplo de IBM MQ. Para utilizar este ejemplo, debe utilizar el archivo ejecutable conectado al cliente, **amqsactc**. El archivo ejecutable se encuentra en el directorio de ejemplos:

- En las plataformas Linux y UNIX, `MQ_INSTALLATION_PATH/samp/bin64`
- En plataformas Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`

Puede utilizar **amqsact** de dos maneras:

Modalidad de visualización

Formatee y visualice mensajes de datos de rastreo de actividad que se están entregando a SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

Modalidad dinámica

Cree una suscripción a un conjunto de recursos y visualice el rastreo de actividad generados ejecutando **amqsact**.

Modalidad de visualización

De forma predeterminada, **amqsact** en modalidad de visualización procesa los mensajes en SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Puede alterar temporalmente este comportamiento especificando un nombre de cola o una serie de tema. El rastreo de actividad debe habilitarse utilizando uno de los métodos que se describen en [Recopilación de información de rastreo de actividad de aplicación](#). Puede controlar el período de rastreo que se visualiza y especifique si los mensajes de rastreo de actividad se eliminan o se conservan después de visualizarse. En modalidad de visualización, **amqsact** toma los argumentos siguientes:

-m nombreGC

Necesario. Especifique el gestor de colas para el que se recopilan los mensajes de rastreo.

-q nombre_cola

Visualice sólo mensajes de rastreo que están relacionados con la cola mencionada.

-t serie_tema

Visualice sólo mensajes de rastreo que están relacionados con el tema mencionado.

-b

Especifique que los mensajes de rastreo se conservan después de visualizarse.

-v

Visualice mensajes de rastreo en modalidad detallada.

-d profundidad

Número de mensajes a visualizar.

-w tiempo_espera

Especifique un tiempo de espera. Si no aparecen mensajes de rastreo en ese período, **amqsact** finaliza.

-s hora_inicio

Utilice este argumento con el argumento **-e** para especificar un periodo de tiempo. Se visualizan mensajes de rastreo del periodo de tiempo especificado.

-e hora_finalización

Utilice este argumento con el argumento **-s** para especificar un periodo de tiempo. Se visualizan mensajes de rastreo del periodo de tiempo especificado.

Por ejemplo, el mandato siguiente muestra los mensajes de rastreo de actividad que se mantienen en SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE y suprime los mensajes después de visualizarlos:

```
amqsact -m QMGR1
```

El mandato siguiente muestra mensajes de rastreo de actividad en la cola especificada, SUB.QUEUE, y suprime los mensajes después de visualizarlos. Los mensajes continúan visualizándose hasta que ha transcurrido un período de 30 segundos sin mensajes nuevos. Por ejemplo, este mandato puede utilizarse con una suscripción a una serie de tema de sistema de rastreo de actividad.

```
amqact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

El mandato siguiente visualiza en formato detallado los datos de rastreo de actividad que se mantienen actualmente en la SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE que se ha producido en el periodo de 20 minutos especificado. Los mensajes permanecerán en la cola después de visualizarse.

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

Modalidad dinámica

La modalidad dinámica se habilita especificando un nombre de aplicación, un nombre de canal o un identificador de conexión como argumento para **amqsact**. Puede utilizar caracteres comodín en el nombre. En modalidad dinámica, los datos de rastreo de actividad se habilitan al iniciar el ejemplo utilizando una suscripción no duradera a un tema del sistema. La recopilación de datos de rastreo de actividad se detiene cuando se detiene **amqsact**. Debe especificar un tiempo de espera para **amqsact** en modalidad dinámica. Puede ejecutar varias copias de **amqsact** simultáneamente y cada instancia recibe una copia de todos los datos de rastreo de actividad. En modalidad dinámica, **amqsact** toma los argumentos siguientes:

-m nombreGC

Necesario. Especifique el gestor de colas para el que se recopilan los mensajes de rastreo.

-w tiempo_espera

Necesario. Especifique un tiempo de espera. Si no aparecen mensajes de rastreo en ese período, **amqsact** finaliza.

-a nombre_aplicación

Especifique una aplicación para la que recopilar mensajes.

-c nombre_canal

Especifique un canal para el que recopilar mensajes.

-i ID_conexión

Especifique una conexión para la que recopilar mensajes.

-v

Visualice mensajes de rastreo en modalidad detallada.

Por ejemplo, el siguiente mandato genera y muestra mensajes de rastreo de actividad para las conexiones realizadas por aplicaciones que se denominan "amqsget.exe". Después de 30 segundos de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

El siguiente mandato genera y muestra mensajes de rastreo de actividad para las conexiones realizadas por aplicaciones que comienzan con el texto "amqs". Después de 30 segundos de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

El siguiente mandato genera y muestra mensajes de rastreo de actividad para cualquier actividad en el canal QMGR1.TO.QMGR2. Después de 10 segundos de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

El siguiente mandato genera y muestra mensajes de rastreo de actividad para cualquier actividad en los canales. Después de 10 segundos de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 10 -c #
```

El siguiente mandato genera y muestra mensajes de rastreo de actividad detallados para cualquier actividad en la conexión de IBM MQ existente que tiene un CONN de "6B576B5420000701" y un EXTCONN de "414D5143514D47523120202020202020". Después de un minuto de inactividad, el programa **amqsact** finaliza, y no se generan nuevos datos de rastreo de actividad.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

Configuración de los niveles de rastreo utilizando mqat.ini

Puede configurar los niveles de rastreo para un gestor de colas estableciendo valores para la stanza AllActivityTrace del archivo de configuración mqat.ini.

Puede establecer los siguientes valores para la stanza AllActivityTrace:

ActivityInterval

Intervalo de tiempo entre mensajes de rastreo. El rastreo de actividad no utiliza una hebra de temporizador, de modo que el mensaje de rastreo no se graba en el instante exacto en que el tiempo transcurre, se graba cuando se ejecuta la primera operación de MQI después de que haya transcurrido el intervalo de tiempo. Si este valor es 0, el mensaje de rastreo se graba cuando la conexión se desconecta (o cuando se alcanza el valor especificado en el parámetro ActivityCount). Toma de forma predeterminada 1.

ActivityCount

Número de operaciones MQI entre mensajes de rastreo. Si este valor es 0, el mensaje de rastreo se graba cuando la conexión se desconecta (o cuando ha transcurrido el intervalo de actividad). Toma de forma predeterminada 100.

TraceLevel

Cantidad de detalle de parámetro que se rastrea para cada operación. La descripción de las operaciones individuales detalla qué parámetros se incluyen para cada nivel de rastreo. Se establece en LOW, MEDIUM o HIGH. Toma de forma predeterminada MEDIUM.

TraceMessageData

Cantidad de datos de mensaje que se rastrean en bytes para operaciones MQGET, MQPUT, MQPUT1 y devolución de llamada. Toma de forma predeterminada 0.

StopOnGetTraceMsg

Se puede establecer en ON u OFF. Toma de forma predeterminada ON.

SubscriptionDelivery

Se puede establecer en BATCHED o IMMEDIATE. Determina si los parámetros ActivityInterval y ActivityCount se utilizarán cuando estén presentes una o más suscripciones de rastreo de actividad. Establecer este parámetro a IMMEDIATE hace que los valores ActivityInterval y ActivityCount se

sustituyan con valores efectivos de 1 cuando los datos de rastreo tienen una suscripción coincidente. Cada registro de rastreo de actividad no se procesa por lotes con otros registros de la misma conexión y en lugar de ello se entrega a la suscripción inmediatamente sin retardo. El valor IMMEDIATE aumenta la sobrecarga de rendimiento de la recopilación de datos de rastreo de actividad. El valor predeterminado es BATCHED.

Referencia de los mensajes de rastreo de actividad de la aplicación

Utilice esta página para obtener una visión general del formato de los mensajes de rastreo de actividad de la aplicación y la información devuelta en estos mensajes

Los mensajes de rastreo de actividad de la aplicación son mensajes IBM MQ estándar que contienen un descriptor de mensaje y datos de mensaje. Los datos de mensaje contienen información sobre las operaciones MQI realizadas por las aplicaciones IBM MQ, o información sobre las actividades que se producen en un sistema IBM MQ.

Descriptor de mensaje

- Una estructura MQMD

Datos de mensaje

- Una cabecera PCF (MQCFH)
- Los datos de mensajes de rastreo de actividad de la aplicación que siempre se devuelven
- Los datos de mensajes de rastreo de actividad de la aplicación que son específicos de la operación

MQMD (descriptor de mensaje) del mensaje de rastreo de actividad de la aplicación

Utilice esta página para comprender las diferencias entre el descriptor de mensaje de los mensajes de rastreo de actividad de la aplicación y el descriptor de mensaje de los mensajes de suceso

Los parámetros y valores del descriptor de mensaje del mensaje de rastreo de actividad de la aplicación son los mismos que en el descriptor de mensaje de mensajes de suceso, con la siguiente excepción:

Format

Descripción:	Nombre de formato de los datos del mensaje.
Valor:	MQFMT_ADMIN Mensaje de administración.

CorrelId

Descripción:	Identificador de correlación.
Valor:	Se inicializa con el ConnectionId de la aplicación

MQCFH (cabecera PCF)

Utilice esta página para ver los valores PCF incluidos en la estructura MQCFH para un mensaje de rastreo de actividad

Para un mensaje de rastreo de actividad, la estructura MQCFH contiene los valores siguientes:

Type

Descripción:	Tipo de estructura que identifica el contenido del mensaje.
Tipo de datos:	MQLONG.
Valor:	MQCFT_APP_ACTIVITY

StrucLength

Descripción:	Longitud en bytes de la estructura MQCFH.
--------------	---

Tipo de datos: MQLONG.
Valor: MQCFH_STRUC_LENGTH

Version

Descripción: Número de versión de la estructura.
Tipo de datos: MQLONG.
Valores: MQCFH_VERSION_3

Command

Descripción: Identificador de mandato. Este campo identifica la categoría del mensaje.
Tipo de datos: MQLONG.
Valores: MQCMD_ACTIVITY_TRACE

MsgSeqNumber

Descripción: Número de secuencia de mensaje. Este campo es el número de secuencia del mensaje dentro de un grupo de mensajes relacionados.
Tipo de datos: MQLONG.
Valores: 1

Control

Descripción: Opciones de control.
Tipo de datos: MQLONG.
Valores: MQCFC_LAST.

CompCode

Descripción: Código de terminación.
Tipo de datos: MQLONG.
Valores: MQCC_OK.

Reason

Descripción: Código de razón que califica el código de terminación.
Tipo de datos: MQLONG.
Valores: MQRC_NONE.

ParameterCount

Descripción: Recuento de las estructuras de parámetro. Este campo es el número de estructuras de parámetro que siguen a la estructura MQCFH. Una estructura de grupo (MQCFGR) y las estructuras de parámetro que incluye, se consideran como una estructura única.
Tipo de datos: MQLONG.
Valores: 1 o más

Datos de mensaje de rastreo de actividad de aplicación

Inmediatamente después de la cabecera PCF hay un conjunto de parámetros que describen el intervalo de tiempo para el rastreo de actividad. Estos parámetros también indican la secuencia de mensajes en el caso de que se graben mensajes. El orden y el número de campos que siguen a la cabecera no está garantizado, lo que permitirá añadir información adicional en el futuro.

Nombre de mensaje:	Mensaje de rastreo de actividad.
Cola de sistema:	SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

QueueManager

Descripción:	Nombre del gestor de colas.
Identificador:	MQCA_Q_MGR_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH

QSGName



Descripción:	El nombre del grupo de compartición de colas del que el gestor de colas es miembro (solo z/OS).
Identificador:	MQCA_QSG_NAME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH

HostName

Descripción:	El nombre de host de la máquina en la que se ejecuta el gestor de colas.
Identificador:	MQCACF_HOST_NAME
Tipo de datos:	MQCFST

IntervalStartDate

Descripción:	La fecha del inicio del periodo de supervisión.
Identificador:	MQCAMO_START_DATE
Tipo de datos:	MQCFST
Longitud máxima:	MQ_DATE_LENGTH

IntervalStartTime

Descripción:	La hora de inicio del periodo de supervisión.
Identificador:	MQCAMO_START_TIME
Tipo de datos:	MQCFST
Longitud máxima:	MQ_TIME_LENGTH

IntervalEndDate

Descripción:	La fecha del final del periodo de supervisión.
Identificador:	MQCAMO_END_DATE
Tipo de datos:	MQCFST

Longitud máxima: MQ_DATE_LENGTH

IntervalEndTime

Descripción: La hora del final del periodo de supervisión.

Identificador: MQCAMO_END_TIME

Tipo de datos: MQCFST

Longitud máxima: MQ_TIME_LENGTH

CommandLevel

Descripción: El nivel de mandato IBM MQ.

Identificador: MQIA_COMMAND_LEVEL

Tipo de datos: MQCFIN

SeqNumber

Descripción: El número de secuencia normalmente es cero. Este valor se incrementa para cada registro subsiguiente para las conexiones de larga ejecución.

Identificador: MQIACF_SEQUENCE_NUMBER

Tipo de datos: MQCFIN

ApplicationName

Descripción: El nombre de la aplicación (nombre de programa).

Identificador: MQCACF_APPL_NAME

Tipo de datos: MQCFST

Longitud máxima: MQ_APPL_NAME_LENGTH

ApplClass

Descripción: Tipo de aplicación que ha realizado la actividad. Valores posibles: MQAT_*

Identificador: MQIA_APPL_TYPE

Tipo de datos: MQCFIN

ApplicationPid

Descripción: El ID de proceso del sistema operativo de la aplicación.

Identificador: MQIACF_PROCESS_ID

Tipo de datos: MQCFIN

UserId

Descripción: El contexto del identificador de usuario de la aplicación.

Identificador: MQCACF_USER_IDENTIFIER

Tipo de datos: MQCFST

Longitud máxima: MQ_USER_ID_LENGTH

APICallerType

Descripción: Tipo de la aplicación. Valores posibles: MQXACT_EXTERNAL o MQXACT_INTERNAL.
Identificador: MQIACF_API_CALLER_TYPE
Tipo de datos: MQCFIN

Environment

Descripción: Entorno de ejecución de la aplicación. Valores posibles: MQXE_*
Identificador: MQIACF_API_ENVIRONMENT
Tipo de datos: MQCFIN

ChannelName

Descripción: El nombre de canal asociado a la conexión. Este parámetro solo se devuelve cuando el valor del parámetro de entorno es MQXE_MCA o MQXE_MCA_SVRCONN.
Identificador: MQCACH_CHANNEL_NAME
Tipo de datos: MQCFST
Longitud máxima: MQ_CHANNEL_NAME_LENGTH

ConnectionName

Descripción: El nombre de conexión de red asociado a la conexión. Este parámetro solo se devuelve cuando el valor del parámetro de entorno es MQXE_MCA o MQXE_MCA_SVRCONN.
Identificador: MQCACH_CONNECTION_NAME
Tipo de datos: MQCFST
Longitud máxima: MQ_CONN_NAME_LENGTH

ChannelType

Descripción: El tipo de canal asociado a la conexión. Este parámetro solo se devuelve cuando el valor del parámetro de entorno es MQXE_MCA o MQXE_MCA_SVRCONN. Valores posibles: MQCHT_*
Identificador: MQIACH_CHANNEL_TYPE
Tipo de datos: MQCFIN

RemoteProduct

Descripción: El identificador de un producto remoto asociado a la conexión. Este parámetro solo se devuelve cuando el valor del parámetro de entorno es MQXE_MCA o MQXE_MCA_SVRCONN.
Identificador: MQCACH_REMOTE_PRODUCT
Tipo de datos: MQCFST
Longitud máxima: MQ_REMOTE_PRODUCT_LENGTH

RemoteVersion

Descripción:	La versión de producto remoto asociada a la conexión. Este parámetro solo se devuelve cuando el valor del parámetro de entorno es MQXE_MCA o MQXE_MCA_SVRCONN.
Identificador:	MQCACH_REMOTE_VERSION
Tipo de datos:	MQCFST
Longitud máxima:	MQ_REMOTE_VERSION_LENGTH

FunctionName

Descripción:	El nombre de la última función de alto nivel iniciada por la hebra inicial.
Identificador:	MQCACF_APPL_FUNCTION
Tipo de datos:	MQCFST

FunctionType

Descripción:	El tipo de la última función de alto nivel iniciada por la hebra inicial. Valores posibles: MQFUN_*
Identificador:	MQIACF_APPL_FUNCTION_TYPE
Tipo de datos:	MQCFIN

Detail

Descripción:	Nivel de detalle que se registra para la conexión. Valores posibles: 1=LOW 2=MEDIUM 3=HIGH
Identificador:	MQIACF_TRACE_DETAIL
Tipo de datos:	MQCFIN

TraceDataLength

Descripción:	La longitud de datos de mensaje (en bytes) que se rastrean para esta conexión.
Identificador:	MQIACF_TRACE_DATA_LENGTH
Tipo de datos:	MQCFIN

PointerSize

Descripción:	La longitud (en bytes) de punteros en la plataforma en la que se está ejecutando la aplicación (para ayudar en la interpretación de estructuras binarias).
Identificador:	MQIACF_POINTER_SIZE
Tipo de datos:	MQCFIN

Platform

Descripción:	La plataforma en la que se ejecuta el gestor de colas. Valores posibles: MQPL_*
Identificador:	MQIA_PLATFORM
Tipo de datos:	MQCFIN

Parámetros variables para las operaciones de MQI de la actividad de la aplicación

La estructura MQCFGR de datos de actividad de la aplicación va seguida del conjunto de parámetros PCF que corresponden a la operación que se está realizando. Los parámetros para cada operación se definen en la siguiente sección.

El nivel de rastreo indica el nivel de granularidad de rastreo que se necesita para los parámetros que se incluirán en el rastreo. Los valores de nivel de rastreo posibles son:

1. Bajo

El parámetro se incluye cuando se configura el rastreo de actividad "low", "medium" o "high" para una aplicación. Este valor significa que siempre se incluye un parámetro en el grupo `AppActivityData` de la operación. Este conjunto de parámetros es suficiente para rastrear las llamadas MQI que una aplicación realiza y para ver si son satisfactorios.

2. Medio

El parámetro sólo se incluye en el grupo `AppActivityData` para la operación cuando se configura el rastreo de actividad "medium" o "high" para una aplicación. Este conjunto de parámetros añade información sobre los recursos, por ejemplo, los nombres de colas y temas utilizados por la aplicación.

3. Alto

El parámetro sólo se incluye en el grupo `AppActivityData` para la operación cuando se configura el rastreo de actividad "high" para una aplicación. Este conjunto de parámetros incluye volcados de memoria de las estructuras que se pasan a las funciones de MQI y XA. Por este motivo, contiene más información sobre los parámetros utilizados en llamadas MQI y XA. Los valores de memoria de la estructura son copias superficiales de las estructuras. Para evitar intentos erróneos de anulación de referencia de los punteros, los valores de puntero en las estructuras se establecen en NULL.

Nota: La versión de la estructura que se vuelca no es necesariamente idéntica a la versión utilizada por una aplicación. La estructura se puede modificar mediante una salida cruzada de API, mediante el código de rastreo de actividad o mediante el gestor de colas. Un gestor de colas puede modificar una estructura en una versión posterior, pero el gestor de colas nunca la cambia a una versión anterior de la estructura. En caso de hacerlo, se corre el riesgo de perder datos.

MQBACK

La aplicación ha iniciado la función MQBACK de MQI

CompCode

Descripción:	Código de terminación que indica el resultado de la operación
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	1
Tipo	MQCFIN

Reason

Descripción:	Código de razón resultado de la operación
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo	MQCFIN

QMGrOpDuration

Descripción:	Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas. La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ. Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF:	MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2
Tipo MQCFIN64

MQBEGIN

La aplicación ha iniciado la función MQBEGIN de MQI

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo MQCFIN

Reason

Descripción: Código de razón resultado de la operación
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 1
Tipo MQCFIN

MQBO

Descripción: La estructura de opciones de MQBEGIN. Este parámetro no se incluye si se utiliza un puntero NULO en la llamada MQBEGIN.
Parámetro PCF: MQBACF_MQBO_STRUCT
Nivel de rastreo: 3
Tipo MQCFBS
Longitud: La longitud en bytes de la estructura MQBO.

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.
Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo: 2
Tipo MQCFIN64

MQCALLBACK

La aplicación ha iniciado la función MQCALLBACK

ObjectHandle

Descripción: Manejador de objeto
Parámetro PCF: MQIACF_HOBJ
Nivel de rastreo: 1
Tipo MQCFIN

CallType

Descripción:	La razón por la que se ha llamado la función. Uno de los valores MQCBCT_*
Parámetro PCF:	MQIACF_CALL_TYPE
Nivel de rastreo:	1
Tipo	MQCFIN

MsgBuffer

Descripción:	Los datos del mensaje.
Parámetro PCF:	MQBACF_MESSAGE_DATA
Nivel de rastreo:	1
Tipo	MQCFBS
Longitud:	La longitud está gobernada por el parámetro TRACEDATA() definido en la configuración de APPTRACE. Si TRACEDATA=NONE, este parámetro se omite.

MsgLength

Descripción:	Longitud del mensaje. (Se toma del campo DataLength de la estructura MQCBC).
Parámetro PCF:	MQIACF_MSG_LENGTH
Nivel de rastreo:	1
Tipo	MQCFIN

HighResTime

Descripción:	Tiempo de la operación, en microsegundos, desde la media noche del 1 de enero de 1970 (UTC). Nota: La precisión de este temporizador varía según el soporte de plataforma para un temporizador de alta resolución
Parámetro PCF:	MQIAMO64_HIGHRES_TIME
Nivel de rastreo:	2
Tipo	MQCFIN64

ReportOptions

Descripción:	Opciones para mensajes de informe
Parámetro PCF:	MQIACF_REPORT
Nivel de rastreo:	2
Tipo	MQCFIN

MsgType

Descripción:	Tipo de mensaje
Parámetro PCF:	MQIACF_MSG_TYPE
Nivel de rastreo:	2
Tipo	MQCFIN

Expiry

Descripción:	Duración del mensaje
Parámetro PCF:	MQIACF_EXPIRY
Nivel de rastreo:	2
Tipo	MQCFIN

Format

Descripción:	Nombre de formato de datos de mensaje
Parámetro PCF:	MQCACH_FORMAT_NAME
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	MQ_FORMAT_LENGTH

Priority

Descripción:	Prioridad de mensaje
Parámetro PCF:	MQIACF_PRIORITY
Nivel de rastreo:	2
Tipo	MQCFIN

Persistence

Descripción:	Persistencia de los mensajes
Parámetro PCF:	MQIACF_PERSISTENCE
Nivel de rastreo:	2
Tipo	MQCFIN

MsgId

Descripción:	Identificador de mensaje
Parámetro PCF:	MQBACF_MSG_ID
Nivel de rastreo:	2
Tipo	MQCFBS
Longitud:	MQ_MSG_ID_LENGTH

CorrelId

Descripción:	Identificador de correlación
Parámetro PCF:	MQBACF_CORREL_ID
Nivel de rastreo:	2
Tipo	MQCFBS
Longitud:	MQ_CORREL_ID_LENGTH

ObjectName

Descripción:	Nombre del objeto abierto.
--------------	----------------------------

Parámetro PCF:	MQCACF_OBJECT_NAME
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	MQ_Q_NAME_LENGTH

ResolvedQName

Descripción:	El nombre local de la cola de la que se ha recuperado el mensaje.
Parámetro PCF:	MQCACF_RESOLVED_Q_NAME
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	MQ_Q_NAME_LENGTH

ReplyToQueue

Descripción:	MQ_Q_NAME_LENGTH
Parámetro PCF:	MQCACF_REPLY_TO_Q
Nivel de rastreo:	2
Tipo	MQCFST

ReplyToQMgr

Descripción:	MQ_Q_MGR_NAME_LENGTH
Parámetro PCF:	MQCACF_REPLY_TO_Q_MGR
Nivel de rastreo:	2
Tipo	MQCFST

CodedCharSetId

Descripción:	Identificador de juego de caracteres de datos de mensaje
Parámetro PCF:	MQIA_CODED_CHAR_SET_ID
Nivel de rastreo:	2
Tipo	MQCFIN

Encoding

Descripción:	Codificación numérica de datos de mensaje.
Parámetro PCF:	MQIACF_ENCODING
Nivel de rastreo:	2
Tipo	MQCFIN

PutDate

Descripción:	MQ_PUT_DATE_LENGTH
Parámetro PCF:	MQCACF_PUT_DATE
Nivel de rastreo:	2
Tipo	MQCFST

PutTime

Descripción:	MQ_PUT_TIME_LENGTH
Parámetro PCF:	MQCACF_PUT_TIME
Nivel de rastreo:	2
Tipo	MQCFST

ResolvedQName

Descripción:	Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	MQ_Q_NAME_LENGTH.

ResObjectString

Descripción:	Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF:	MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	La longitud varía.

ResolvedType

Descripción:	Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF:	MQIACF_RESOLVED_TYPE
Nivel de rastreo:	2
Tipo	MQCFIN

PolicyName

Descripción:	Nombre de la política aplicada al mensaje. Nota: Sólo para mensajes protegidos por AMS.
Parámetro PCF:	MQCA_POLICY_NAME
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	MQ_OBJECT_NAME_LENGTH

XmitqMsgId

Descripción:	ID del mensaje en la cabecera de la cola de transmisión. Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.
Parámetro PCF:	MQBACF_XQH_MSG_ID

Nivel de rastreo: 2
Tipo MQCFBS
Longitud: MQ_MSG_ID_LENGTH

XmitqCorrelId

Descripción: ID de correlación del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQBACF_XQH_CORREL_ID
Nivel de rastreo: 2
Tipo MQCFBS
Longitud: MQ_CORREL_ID_LENGTH

XmitqPutTime

Descripción: Hora de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_TIME
Nivel de rastreo: 2
Tipo MQCFST
Longitud: MQ_PUT_TIME_LENGTH

XmitqPutDate

Descripción: Fecha de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_DATE
Nivel de rastreo: 2
Tipo MQCFST
Longitud: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_Name
Nivel de rastreo: 2
Tipo MQCFST
Longitud: MQ_Q_NAME_LENGTH

XmitqRemoteQMgr

Descripción: ID del mensaje en la cabecera de la cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_MGR

Nivel de rastreo: 2
Tipo MQCFST
Longitud: MQ_MSG_ID_LENGTH

MsgDescStructure

Descripción: Estructura MQMD. Este parámetro se omite si se utilizó una versión 4 MQGMO para solicitar que se devuelva un manejador de mensajes en lugar de un MQMD
Parámetro PCF: MQBACF_MQMD_STRUCT
Nivel de rastreo: 3
Tipo MQCFBS
Longitud: La longitud en bytes de la estructura MQMD (el tamaño real depende de la versión de la estructura)

GetMsgOptsStructure

Descripción: Estructura MQGMO.
Parámetro PCF: MQBACF_MQGMO_STRUCT
Nivel de rastreo: 3
Tipo MQCFBS
Longitud: La longitud en bytes de la estructura MQGMO (el tamaño real depende de la versión de la estructura)

MQCBCContextStructure

Descripción: La estructura MQCBC.
Parámetro PCF: MQBACF_MQCBC_STRUCT
Nivel de rastreo: 3
Tipo MQCFBS
Longitud: La longitud en bytes de la estructura MQCBC (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.
La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.
Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF: MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo: 2
Tipo MQCFIN64

MQCB

La aplicación ha iniciado la función de gestión de devolución de llamada de MQI

CallbackOperation

Descripción:	La operación de función de gestión de devolución de llamada. Se establece en uno de los valores MQOP_*
Parámetro PCF:	MQIACF_MQCB_OPERATION
Nivel de rastreo:	1
Tipo	MQCFIN

CallbackType

Descripción:	Tipo de la función de devolución de llamada (campo CallbackType de la estructura MQCBD). Se establece en uno de los valores MQCBT_*
Parámetro PCF:	MQIACF_MQCB_TYPE
Nivel de rastreo:	1
Tipo	MQCFIN

CallbackOptions

Descripción:	Las opciones de devolución de llamada. Se establece en uno de los valores MQCBDO_*
Parámetro PCF:	MQIACF_MQCB_OPTIONS
Nivel de rastreo:	1
Tipo	MQCFIN

CallbackFunction

Descripción:	Puntero hacia la función de devolución de llamada si se ha iniciado como una llamada a función.
Parámetro PCF:	MQBACF_MQCB_FUNCTION
Nivel de rastreo:	1
Tipo	MQCFBS
Longitud:	Tamaño de MQPTR

CallbackName

Descripción:	Nombre de la función de devolución de llamada si se ha iniciado como programa enlazado dinámicamente.
Parámetro PCF:	MQCACF_MQCB_NAME
Nivel de rastreo:	1
Tipo	MQCFST
Longitud:	Tamaño de MQCHAR128

ObjectHandle

Descripción:	Manejador de objeto
Parámetro PCF:	MQIACF_HOBJ
Nivel de rastreo:	1
Tipo	MQCFIN

MaxMsgLength

Descripción:	La longitud máxima de mensajes. Se establece en un entero o el valor especial MQCBD_FULL_MSG_LENGTH
Parámetro PCF:	MQIACH_MAX_MSG_LENGTH
Nivel de rastreo:	2
Tipo	MQCFIN

CompCode

Descripción:	Código de terminación que indica el resultado de la operación
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	1
Tipo	MQCFIN

Reason

Descripción:	Código de razón resultado de la operación
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo	MQCFIN

ResolvedQName

Descripción:	Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	MQ_Q_NAME_LENGTH.

ResObjectString

Descripción:	Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF:	MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo:	2
Tipo	MQCFST
Longitud:	La longitud varía.

ResolvedType

Descripción:	Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF:	MQIACF_RESOLVED_TYPE
Nivel de rastreo:	2
Tipo	MQCFIN

Callback DescriptorStructure

Descripción:	La estructura MQCBD. Este parámetro se omite si se pasa un valor NULL MQCBC a la llamada MQCB.
Parámetro PCF:	MQBACF_MQCBD_STRUCT
Nivel de rastreo:	3
Tipo	MQCFBS
Longitud:	La longitud en bytes de la estructura MQCBC.

MsgDescStructure

Descripción:	Estructura MQMD. El parámetro MsgDescStructure se omite si se pasa un valor NULL MQCBC a la llamada MQCB.
Parámetro PCF:	MQBACF_MQMD_STRUCT
Nivel de rastreo:	3
Tipo	MQCFBS
Longitud:	La longitud en bytes de la estructura MQMD (el tamaño real depende de la versión de la estructura)

GetMsgOptsStructure

Descripción:	Estructura MQGMO. Este parámetro se omite si se pasa un valor NULL MQGMO a la llamada MQCB.
Parámetro PCF:	MQBACF_MQGMO_STRUCT
Nivel de rastreo:	3
Tipo	MQCFBS
Longitud:	La longitud en bytes de la estructura MQGMO (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción:	Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas. La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ. Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF:	MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo:	2
Tipo	MQCFIN64

MQCLOSE

La aplicación ha iniciado la función MQCLOSE de MQI

ObjectHandle

Descripción:	Manejador de objeto
Parámetro PCF:	MQIACF_HOBJ
Nivel de rastreo:	1

Tipo MQCFIN

CloseOptions

Descripción: Opciones de cierre
Parámetro PCF: MQIACF_CLOSE_OPTIONS
Nivel de rastreo: 1
Tipo MQCFIN

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo MQCFIN

Reason

Descripción: Código de razón resultado de la operación
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 1
Tipo MQCFIN

ResolvedQName

Descripción: Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo: 2
Tipo MQCFST
Longitud: MQ_Q_NAME_LENGTH.

ResObjectString

Descripción: Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo: 2
Tipo MQCFST
Longitud: La longitud varía.

ResolvedType

Descripción: Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF: MQIACF_RESOLVED_TYPE
Nivel de rastreo: 2
Tipo MQCFIN

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2

Tipo MQCFIN64

MQCMIT

La aplicación ha iniciado la función MQCMIT de MQI

CompCode

Descripción: Código de terminación que indica el resultado de la operación

Parámetro PCF: MQIACF_COMP_CODE

Nivel de rastreo: 1

Tipo MQCFIN

Reason

Descripción: Código de razón resultado de la operación

Parámetro PCF: MQIACF_REASON_CODE

Nivel de rastreo: 1

Tipo MQCFIN

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2

Tipo MQCFIN64

MQCONN y MQCONNX

La aplicación ha iniciado la función MQCONN o MQCONNX de MQI

ConnectionId

Descripción: ID de conexión si está disponible, o QCONNID_NONE si no lo está

Parámetro PCF: MQBACF_CONNECTION_ID

Nivel de rastreo: 1

Tipo: MQCFBS

Longitud máxima: MQ_CONNECTION_ID_LENGTH

QueueManagerName

Descripción: Nombre (no resuelto) del gestor de colas utilizado en la llamada MQCONN(X)

Parámetro PCF: MQCA_Q_MGR_NAME

Nivel de rastreo: 1

Tipo: MQCFST

Longitud máxima: MQ_Q_MGR_NAME_LENGTH

CompCode

Descripción: Código de terminación que indica el resultado de la operación

Parámetro PCF: MQIACF_COMP_CODE

Nivel de rastreo: 1

Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación

Parámetro PCF: MQIACF_REASON_CODE

Nivel de rastreo: 1

Tipo: MQCFIN

ConnectOptions

Descripción: Las opciones de conexión derivadas de los valores MQCNO_*

Nota: Sólo para MQCONNX

Parámetro PCF: MQIACF_CONNECT_OPTIONS

Nivel de rastreo: 2

Tipo: MQCFIN

ConnectionOptionsStructure

Descripción: La estructura MQCNO.

Nota: Sólo para MQCONNX)

Parámetro PCF: MQBACF_MQCNO_STRUCT

Nivel de rastreo: 3

Tipo: MQCFBS

Longitud máxima: La longitud en bytes de la estructura MQCNO (el tamaño real depende de la versión de la estructura)

ChannelDefinitionStructure

Descripción: La estructura MQCD.

Nota: Sólo para conexiones de clientes

Parámetro PCF: MQBACF_MQCD_STRUCT

Nivel de rastreo:	3
Tipo:	MQCFBS
Longitud máxima:	La longitud en bytes de la estructura MQCD (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2

Tipo: MQCFIN64

MQCTL

La aplicación ha iniciado la función MQCTL de MQI

CompCode

Descripción: Código de terminación que indica el resultado de la operación

Parámetro PCF: MQIACF_COMP_CODE

Nivel de rastreo: 1

Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación

Parámetro PCF: MQIACF_REASON_CODE

Nivel de rastreo: 1

Tipo: MQCFIN

CtlOperation

Descripción: Uno de los valores MQOP_*

Parámetro PCF: MQIACF_CTL_OPERATION

Nivel de rastreo: 1

Tipo: MQCFIN

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo: 2
Tipo MQCFIN64

MQDISC

La aplicación ha iniciado la función MQDISC de MQI

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

MQGET

La aplicación ha iniciado la función MQGET de MQI

ObjectHandle

Descripción: Manejador de objeto
Parámetro PCF: MQIACF_HOBJ
Nivel de rastreo: 1
Tipo: MQCFIN

GetOptions

Descripción: Las opciones get de MQGMO.Options
Parámetro PCF: MQIACF_GET_OPTIONS
Nivel de rastreo: 1
Tipo: MQCFIN

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 1

Tipo: MQCFIN

MsgBuffer

Descripción: Los datos del mensaje. Si TRACEDATA=NONE, este parámetro se omitirá

Parámetro PCF: MQBACF_MESSAGE_DATA

Nivel de rastreo: 1

Tipo: MQCFBS

Longitud máxima: La longitud está gobernada por el parámetro TRACEDATA() definido en la configuración de APPTRACE. (Se incluye en el mensaje de rastreo como MQIACF_TRACE_DATA_LENGTH).

MsgLength

Descripción: Longitud del mensaje.

Parámetro PCF: MQIACF_MSG_LENGTH

Nivel de rastreo: 1

Tipo: MQCFIN

HighResTime

Descripción: Tiempo de operación en microsegundos desde la medianoche del 1 de enero de 1970 (hora UTC)

Nota: La precisión de este temporizador varía según el soporte de plataforma para un temporizador de alta resolución

Parámetro PCF: MQIAMO64_HIGHRES_TIME

Nivel de rastreo: 2

Tipo: MQCFIN64

BufferLength

Descripción: Longitud del almacenamiento intermedio proporcionado por la aplicación

Parámetro PCF: MQIACF_BUFFER_LENGTH

Nivel de rastreo: 2

Tipo: MQCFIN

ObjectName

Descripción: Nombre del objeto abierto

Parámetro PCF: MQCACF_OBJECT_NAME

Nivel de rastreo: 2

Tipo: MQCFST

Longitud: MQ_Q_NAME_LENGTH

ResolvedQName

Descripción: El nombre local de la cola de la que se ha recuperado el mensaje.

Parámetro PCF: MQCACF_RESOLVED_Q_NAME

Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_Q_NAME_LENGTH

ReportOptions

Descripción: Opciones de informe de mensaje
Parámetro PCF: MQIACF_REPORT
Nivel de rastreo: 2
Tipo: MQCFIN

MsgType

Descripción: Tipo de mensaje
Parámetro PCF: MQIACF_MSG_TYPE
Nivel de rastreo: 2
Tipo: MQCFIN

Expiry

Descripción: Duración del mensaje
Parámetro PCF: MQIACF_EXPIRY
Nivel de rastreo: 2
Tipo: MQCFIN

Format

Descripción: Nombre de formato de datos de mensaje
Parámetro PCF: MQCACH_FORMAT_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_FORMAT_LENGTH

Priority

Descripción: Prioridad de mensaje
Parámetro PCF: MQIACF_PRIORITY
Nivel de rastreo: 2
Tipo: MQCFIN

Persistence

Descripción: Persistencia de los mensajes
Parámetro PCF: MQIACF_PERSISTENCE
Nivel de rastreo: 2
Tipo: MQCFIN

MsgId

Descripción: Identificador de mensaje
Parámetro PCF: MQBACF_MSG_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud máxima: MQ_MSG_ID_LENGTH

CorrelId

Descripción: Identificador de correlación
Parámetro PCF: MQBACF_CORREL_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud máxima: MQ_CORREL_ID_LENGTH

ReplyToQueue

Descripción:
Parámetro PCF: MQCACF_REPLY_TO_Q
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_Q_NAME_LENGTH

ReplyToQMgr

Descripción:
Parámetro PCF: MQCACF_REPLY_TO_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_Q_MGR_NAME_LENGTH

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensaje
Parámetro PCF: MQIA_CODED_CHAR_SET_ID
Nivel de rastreo: 2
Tipo: MQCFIN

Encoding

Descripción: Codificación numérica de datos de mensaje.
Parámetro PCF: MQIACF_ENCODING
Nivel de rastreo: 2
Tipo: MQCFIN

PutDate

Descripción:
Parámetro PCF: MQCACF_PUT_DATE
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_PUT_DATE_LENGTH

PutTime

Descripción:
Parámetro PCF: MQCACF_PUT_TIME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_PUT_TIME_LENGTH

ResolvedQName

Descripción: Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH.

ResObjectString

Descripción: Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: La longitud varía.

ResolvedType

Descripción: Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF: MQIACF_RESOLVED_TYPE
Nivel de rastreo: 2
Tipo: MQCFIN

PolicyName

Descripción: Nombre de la política aplicada al mensaje.
Nota: Sólo para mensajes protegidos por AMS.
Parámetro PCF: MQCA_POLICY_NAME
Nivel de rastreo: 2

Tipo: MQCFST
Longitud: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

Descripción: ID del mensaje en la cabecera de la cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQBACF_XQH_MSG_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_MSG_ID_LENGTH

XmitqCorrelId

Descripción: ID de correlación del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQBACF_XQH_CORREL_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_CORREL_ID_LENGTH

XmitqPutTime

Descripción: Hora de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_TIME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_TIME_LENGTH

XmitqPutDate

Descripción: Fecha de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_DATE
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_NAME
Nivel de rastreo: 2

Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

XmitqRemoteQMGr

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

MsgDescStructure

Descripción: Estructura MQMD.
Parámetro PCF: MQBACF_MQMD_STRUCT
Nivel de rastreo: 3
Tipo: MQCFBS
Longitud máxima: La longitud en bytes de la estructura MQMD (el tamaño real depende de la versión de la estructura)

GetMsgOptsStructure

Descripción: Estructura MQGMO.
Parámetro PCF: MQBACF_MQGMO_STRUCT
Nivel de rastreo: 3
Tipo: MQCFBS
Longitud máxima: La longitud en bytes de la estructura MQGMO (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.
La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.
Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF: MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo: 2
Tipo: MQCFIN64

MQINQ

La aplicación ha iniciado la función MQINQ de MQI

ObjectHandle

Descripción: Manejador de objeto

Parámetro PCF: MQIACF_HOBJ
Nivel de rastreo: 1
Tipo: MQCFIN

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

SelectorCount

Descripción: La cuenta de selectores que se suministran en la matriz Selectors.
Parámetro PCF: MQIACF_SELECTOR_COUNT
Nivel de rastreo: 2
Tipo: MQCFIN

Selectors

Descripción: La lista de atributos (entero o carácter) cuyos valores deben devolverse MQINQ.
Parámetro PCF: MQIACF_SELECTORS
Nivel de rastreo: 2
Tipo: MQCFIL

ResolvedQName

Descripción: Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF: MQCACF_RESOLVED_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_Q_NAME_LENGTH

ResObjectString

Descripción: Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo: 2
Tipo: MQCFST

Longitud máxima: La longitud varía

ResolvedType

Descripción: Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.

Parámetro PCF: MQIACF_RESOLVED_TYPE

Nivel de rastreo: 2

Tipo: MQCFIN

IntAttrCount

Descripción: Número de atributos de tipo entero que devuelve la operación inquire (consultar)

Parámetro PCF: MQIACF_INTATTR_COUNT

Nivel de rastreo: 3

Tipo: MQCFIN

IntAttr

Descripción: Los valores de atributos de tipo entero que devuelve la operación inquire (consultar). Este parámetro solo está presente si IntAttrCount es > 0 cuando se devuelve MQINQ.

Parámetro PCF: MQIACF_INT_ATTRS

Nivel de rastreo: 3

Tipo: MQCFIL

CharAttr

Descripción: Los atributos de tipo carácter que devuelve la operación inquire (consultar). Los valores se concatenan entre sí. Este parámetro solo se incluye si CharAttrLength es > 0 cuando se devuelve MQINQ.

Parámetro PCF: MQCACF_CHAR_ATTRS

Nivel de rastreo: 3

Tipo: MQCFST

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2

Tipo: MQCFIN64

MQOPEN

La aplicación ha iniciado la función MQOPEN de MQI

ObjectType

Descripción:	Tipo de objeto transferido en MQOT.ObjectType
Parámetro PCF:	MQIACF_OBJECT_TYPE
Nivel de rastreo:	1
Tipo:	MQCFIN

ObjectName

Descripción:	Nombre del objeto pasado a la llamada MQI antes de que se haya intentado resolver algún nombre de cola.
Parámetro PCF:	MQCACF_OBJECT_NAME
Nivel de rastreo:	1
Tipo:	MQCFST
Longitud máxima:	MQ_Q_NAME_LENGTH

ObjectQMgrName

Descripción:	Nombre del gestor de colas del objeto transferido a la llamada MQI antes de intentar cualquier resolución de nombre de cola.
Parámetro PCF:	MQCACF_OBJECT_Q_MGR_NAME
Nivel de rastreo:	1
Tipo:	MQCFST
Longitud máxima:	MQ_Q_MGR_NAME_LENGTH

ObjectHandle

Descripción:	Manejador de objeto
Parámetro PCF:	MQIACF_HOBJ
Nivel de rastreo:	1
Tipo:	MQCFIN

CompCode

Descripción:	Código de terminación que indica el resultado de la operación
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

Reason

Descripción:	Código de razón resultado de la operación
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

OpenOptions

Descripción:	Opciones utilizadas para abrir el objeto
--------------	--

Parámetro PCF: MQIACF_OPEN_OPTIONS
Nivel de rastreo: 1
Tipo: MQCFIN

AlternateUserId

Descripción: Sólo se incluye si se especifica MQOO_ALTERNATE_USER_AUTHORITY
Parámetro PCF: MQCACF_ALTERNATE_USERID
Nivel de rastreo: 2
Tipo: MQCFST
Longitud máxima: MQ_USER_ID_LENGTH

RecsPresent

Descripción: Número de registros de nombre de objeto existentes. Solo se incluye si la versión de MQOD es > = MQOD_VERSION_2
Parámetro PCF: MQIACF_RECS_PRESENT
Nivel de rastreo: 1
Tipo: MQCFIN

KnownDestCount

Descripción: Número de colas locales abiertas satisfactoriamente. Solo se incluye si la versión de MQOD es > = MQOD_VERSION_2
Parámetro PCF: MQIACF_KNOWN_DEST_COUNT
Nivel de rastreo: 1
Tipo: MQCFIN

UnknownDestCount

Descripción: Número de colas remotas abiertas satisfactoriamente. Solo se incluye si la versión de MQOD es > = MQOD_VERSION_2
Parámetro PCF: MQIACF_UNKNOWN_DEST_COUNT
Nivel de rastreo: 1
Tipo: MQCFIN

InvalidDestCount

Descripción: Número de colas que no se han podido abrir. Solo se incluye si la versión de MQOD es > = MQOD_VERSION_2
Parámetro PCF: MQIACF_INVALID_DEST_COUNT
Nivel de rastreo: 1
Tipo: MQCFIN

DynamicQName

Descripción: Nombre de cola dinámica transferido como entrada a la llamada MQOPEN.
Parámetro PCF: MQCACF_DYNAMIC_Q_NAME
Nivel de rastreo: 2

Tipo: MQCFST
Longitud máxima: MQ_Q_NAME_LENGTH

ResolvedLocalQName ^{1 2}

Descripción: Contiene el nombre de cola local después de que se ha llevado a cabo la resolución de nombres. (por ejemplo, para colas remotas, será el nombre de la cola de transmisión)

Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME

Nivel de rastreo: 2

Tipo: MQCFST

Rango: Si MQOD.Version es menor que MQOD_VERSION_3, contiene el valor del campo MQOD.ObjectName después de que haya finalizado la llamada MQOPEN. Si MQOD.Version es igual o mayor que MQOD_VERSION_3, contiene el valor del campo MQOD.ResolvedQName.

Longitud máxima: MQ_Q_NAME_LENGTH

ResolvedLocalQMgrName ^{1 2}

Descripción: Nombre del gestor de colas local después de que se haya realizado la resolución de nombres.

Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_MGR

Nivel de rastreo: 2

Tipo: MQCFST

Rango: Solo si MQOD.Version >= MQOD_VERSION_3

Longitud máxima: MQ_Q_MGR_NAME_LENGTH

ResolvedQName ^{1 2}

Descripción: Nombre de cola una vez que se haya llevado a cabo la resolución de nombres.

Parámetro PCF: MQCACF_RESOLVED_Q_NAME

Nivel de rastreo: 2

Tipo: MQCFST

Rango: Si MQOD.Version es menor que MQOD_VERSION_3, contiene el valor del campo MQOD.ObjectName después de que haya finalizado la llamada MQOPEN. Si MQOD.Version es igual o mayor que MQOD_VERSION_3, contiene el valor del campo MQOD.ResolvedQName.

Longitud máxima: MQ_Q_NAME_LENGTH

ResolvedQMgrName ^{1 2}

Descripción: Contiene el nombre del gestor de colas una vez que se ha llevado a cabo la resolución de nombres. Si MQOD.Version es menor que MQOD_VERSION_3, esto contiene el valor de MQOD. El campo NombreGestColasObjetos una vez que se ha completado la llamada MQOPEN. Si MQOD.Version es igual o mayor que MQOD_VERSION_3, contiene el valor del campo MQOD. Campo ResolvedQMgrName.

Parámetro PCF: MQCACF_RESOLVED_Q_MGR

Nivel de rastreo: 2

Tipo: MQCFST
Longitud máxima: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

Descripción: Identificador de seguridad alternativo. Sólo está presente si MQOD.Version es igual o mayor que MOOD_VERSION_3, se especifica MQOO_ALTERNATE_USER_AUTHORITY y MQOD.AlternateSecurityId no es igual a MQSID_NONE.

Parámetro PCF: MQBACF_ALTERNATE_SECURITYID

Nivel de rastreo: 2

Tipo: MQCFBS

Longitud máxima: MQ_SECURITY_ID_LENGTH

ObjectString

Descripción: Nombre de objeto largo. Sólo se incluye si MQOD.Version es igual o mayor que MQOD_VERSION_4 y el campo VSLength de MQOD.ObjectString es MQVS_NULL_TERMINATED o mayor que cero.

Parámetro PCF: MQCACF_OBJECT_STRING

Nivel de rastreo: 2

Tipo: MQCFST

Longitud máxima: La longitud varía.

SelectionString

Descripción: Serie de selección. Sólo se incluye si MQOD.Version es igual o mayor que MQOD_VERSION_4 y el campo VSLength de MQOD.SelectionString es MQVS_NULL_TERMINATED o mayor que cero.

Parámetro PCF: MQCACF_SELECTION_STRING

Nivel de rastreo: 2

Tipo: MQCFST

Longitud máxima: La longitud varía.

ResObjectString

Descripción: Nombre de objeto largo después de que el gestor de colas resuelva el nombre proporcionado en el campo ObjectName. Sólo se incluye para los temas y los alias de colas que hacen referencia a un objeto de tema si MQOD.Version es igual o mayor que MQOD_VERSION_4 y VSLength es MQVS_NULL_TERMINATED o mayor que cero.

Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING

Nivel de rastreo: 2

Tipo: MQCFST

Longitud máxima: La longitud varía.

ResolvedType

Descripción:	El tipo del objeto (base) resuelto que se está abriendo. Sólo se incluye si MQOD.Version es igual o mayor que MQOD_VERSION_4. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF:	MQIACF_RESOLVED_TYPE
Nivel de rastreo:	2
Tipo:	MQCFIN

QMGrOpDuration

Descripción:	Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas. La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ. Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF:	MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo:	2
Tipo:	MQCFIN64

Estructura de cabecera del grupo PCF de la lista de distribución de actividad de aplicación

Si la función MQOPEN abre una lista de distribución, los parámetros MQOPEN incluyen un grupo PCF AppActivityDistList para cada una de las colas de la lista de distribución hasta el número de estructuras numeradas en RecsPresent. El grupo PCF AppActivityDistList combina información de las estructuras MQOR Y MQRR para identificar el nombre de cola e indicar el resultado de la operación de apertura en la cola. Un grupo AppActivityDistList siempre empieza con la siguiente estructura MQCFGR:

Campo MQCFGR	Valor	Descripción
Tipo	MQCFT_GROUP	
StrucLength	Longitud en bytes de la estructura MQCFGR	
Parámetro	MQGACF_APP_DIST_LIST	Parámetro de grupo de lista de distribución
ParameterCount	4	El número de estructuras de parámetro que siguen a la estructura MQCFGR que están dentro de este grupo.

ObjectName

Descripción:	El nombre de una cola en la lista de distribución MQ_Q_NAME_LENGTH. Sólo se incluye si se proporcionan estructuras MQOR.
Parámetro PCF:	MQCACF_OBJECT_NAME

¹ Este parámetro sólo se incluye si el objeto que se abre se resuelve en una cola y la cola se abre para MQOO_INPUT_*, MQOO_OUTPUT o MQOO_BROWSE

² El parámetro ResolvedLocalQName sólo se incluye si es distinto del parámetro ResolvedQName.

Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH. Sólo se incluye si se proporcionan estructuras MQOR.

ObjectQMgrName

Descripción: El nombre del gestor de colas en el que se define la cola nombrada en ObjectName.
Parámetro PCF: MQCACF_OBJECT_Q_MGR_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_MGR_NAME_LENGTH. Sólo se incluye si se proporcionan estructuras MQOR.

CompCode

Descripción: El código de terminación que indica el resultado de la apertura para este objeto. Sólo se incluye si se proporcionan estructuras MQRR y el código de razón para MQOPEN es MQRC_MULTIPLE_REASONS
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 2
Tipo: MQCFIN

Reason

Descripción: El código de razón que indica el resultado de la apertura para este objeto. Sólo se incluye si se proporcionan estructuras MQRR y el código de razón para MQOPEN es MQRC_MULTIPLE_REASONS
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 2
Tipo: MQCFIN

MQPUT

La aplicación ha iniciado la función MQPUT MQI.

ObjectHandle

Descripción: Manejador de objeto
Parámetro PCF: MQIACF_HOBJ
Nivel de rastreo: 1
Tipo: MQCFIN

PutOptions

Descripción: Opciones de transferencia de MQPMO.Options
Parámetro PCF: MQIACF_PUT_OPTIONS
Nivel de rastreo: 1
Tipo: MQCFIN

CompCode

Descripción:	Código de terminación que indica el resultado de la operación
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

Reason

Descripción:	Código de razón resultado de la operación
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

MsgBuffer

Descripción:	Los datos del mensaje.
Parámetro PCF:	MQBACF_MESSAGE_DATA
Nivel de rastreo:	1
Tipo:	MQCFBS
Longitud:	La longitud está gobernada por el parámetro TRACEDATA() definido en la configuración de APPTRACE. Si TRACEDATA=NONE, este parámetro se omite.

MsgLength

Descripción:	Longitud del mensaje.
Parámetro PCF:	MQIACF_MSG_LENGTH
Nivel de rastreo:	1
Tipo:	MQCFIN

RecsPresent

Descripción:	Número de registros de mensajes de transferencia o registros de respuesta presentes. Solo se incluye si la versión de MQPMO es \geq MQPMO_VERSION_2
Parámetro PCF:	MQIACF_RECS_PRESENT
Nivel de rastreo:	1
Tipo:	MQCFIN

KnownDestCount

Descripción:	Número de mensajes enviados satisfactoriamente a las colas locales.
Parámetro PCF:	MQIACF_KNOWN_DEST_COUNT
Nivel de rastreo:	1
Tipo:	MQCFIN

UnknownDestCount

Descripción:	Número de mensajes enviados satisfactoriamente a las colas remotas.
Parámetro PCF:	MQIACF_UNKNOWN_DEST_COUNT

Nivel de rastreo: 1
Tipo: MQCFIN

InvalidDestCount

Descripción: Número de mensajes que no se han podido enviar
Parámetro PCF: MQIACF_INVALID_DEST_COUNT
Nivel de rastreo: 1
Tipo: MQCFIN

HighResTime

Descripción: Tiempo de la operación, en microsegundos, desde la media noche del 1 de enero de 1970 (UTC).
Nota: La precisión de este temporizador varía según el soporte de plataforma para un temporizador de alta resolución.
Parámetro PCF: MQIAMO64_HIGHRES_TIME
Nivel de rastreo: 2
Tipo: MQCFIN64

ObjectName

Descripción: Nombre del objeto abierto.
Parámetro PCF: MQCACF_OBJECT_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

ResolvedQName

Descripción: Nombre de la cola después de haber realizar la resolución de nombre de cola.
Parámetro PCF: MQCACF_RESOLVED_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

ResolvedQMgrName

Descripción: Nombre del gestor de colas después de haber realizado la resolución de nombre.
Parámetro PCF: MQCACF_RESOLVED_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_MGR_NAME_LENGTH

ResolvedLocalQName ³

Descripción:	Contiene el nombre de cola local después de que se ha llevado a cabo la resolución de nombres.
Parámetro PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo:	2
Tipo:	MQCFST

ResolvedLocalQMgrName ³

Descripción:	Contiene el nombre de gestor de colas local después de haber realizado la resolución de nombre.
Parámetro PCF:	MQCACF_RESOLVED_LOCAL_Q_MGR
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_Q_MGR_NAME_LENGTH

ReportOptions

Descripción:	Opciones de informe de mensaje
Parámetro PCF:	MQIACF_REPORT
Nivel de rastreo:	2
Tipo:	MQCFIN

MsgType

Descripción:	Tipo de mensaje
Parámetro PCF:	MQIACF_MSG_TYPE
Nivel de rastreo:	2
Tipo:	MQCFIN

Expiry

Descripción:	Duración del mensaje
Parámetro PCF:	MQIACF_EXPIRY
Nivel de rastreo:	2
Tipo:	MQCFIN

Format

Descripción:	Nombre de formato de datos de mensaje
Parámetro PCF:	MQCACH_FORMAT_NAME
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_FORMAT_LENGTH

Priority

Descripción:	Prioridad de mensaje
--------------	----------------------

Parámetro PCF: MQIACF_PRIORITY
Nivel de rastreo: 2
Tipo: MQCFIN

Persistence

Descripción: Persistencia de los mensajes
Parámetro PCF: MQIACF_PERSISTENCE
Nivel de rastreo: 2
Tipo: MQCFIN

MsgId

Descripción: Identificador de mensaje
Parámetro PCF: MQBACF_MSG_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_MSG_ID_LENGTH

CorrelId

Descripción: Identificador de correlación
Parámetro PCF: MQBACF_CORREL_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_CORREL_ID_LENGTH

ReplyToQueue

Descripción:
Parámetro PCF: MQCACF_REPLY_TO_Q
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

ReplyToQMgr

Descripción:
Parámetro PCF: MQCACF_REPLY_TO_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_MGR_NAME_LENGTH

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensaje
Parámetro PCF: MQIA_CODED_CHAR_SET_ID

Nivel de rastreo: 2
Tipo: MQCFIN

Encoding

Descripción: Codificación numérica de datos de mensaje.
Parámetro PCF: MQIACF_ENCODING
Nivel de rastreo: 2
Tipo: MQCFIN

PutDate

Descripción:
Parámetro PCF: MQCACF_PUT_DATE
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_DATE_LENGTH

PutTime

Descripción:
Parámetro PCF: MQCACF_PUT_TIME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_TIME_LENGTH

ResolvedQName

Descripción: Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH.

ResObjectString

Descripción: Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: La longitud varía.

ResolvedType

Descripción: Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.

Parámetro PCF: MQIACF_RESOLVED_TYPE
Nivel de rastreo: 2
Tipo: MQCFIN

PolicyName

Descripción: Nombre de la política aplicada al mensaje.
Nota: Sólo para mensajes protegidos por AMS.

Parámetro PCF: MQCA_POLICY_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

Descripción: ID del mensaje en la cabecera de la cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQBACF_XQH_MSG_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_MSG_ID_LENGTH

XmitqCorrelId

Descripción: ID de correlación del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQBACF_XQH_CORREL_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_CORREL_ID_LENGTH

XmitqPutTime

Descripción: Hora de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_TIME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_TIME_LENGTH

XmitqPutDate

Descripción: Fecha de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_DATE

Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.

Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

XmitqRemoteQMGr

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.

Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

PutMsgOptsStructure

Descripción: Estructura MQPMO.
Parámetro PCF: MQBACF_MQPMO_STRUCT
Nivel de rastreo: 3
Tipo: MQCFBS
Longitud: Longitud en bytes de la estructura MQPMO (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo: 2
Tipo: MQCFIN64

³ El parámetro ResolvedLocalQName sólo se incluye si es distinto del parámetro ResolvedQName.

Estructura de cabecera de grupo PCF de lista de distribución de actividad de aplicación de MQPUT

Si la función MQPUT transfiere datos a una lista de distribución, los parámetros de MQPUT incluyen un grupo PCF AppActivityDistList. Para cada una de las colas de la lista de distribución, consulte [“Estructura de cabecera del grupo PCF de la lista de distribución de actividad de aplicación”](#) en la página 259. El grupo PCF AppActivityDistList combina información de las estructuras MQPMR y MQRR para identificar los parámetros PUT e indica el resultado de la operación PUT en cada cola. Para operaciones MQPUT, el grupo AppActivityDistList contiene algunos o todos los parámetros siguientes (CompCode y Reason están presentes si el código de razón es MQRC_MULTIPLE_REASONS y los otros parámetros se determinan mediante el campo MQPMO.PutMsgRecFields):

CompCode

Descripción:	Código de terminación que indica el resultado de la operación. Sólo se incluye si se proporcionan estructuras MQRR y si el código de razón para MQPUT es MQRC_MULTIPLE_REASONS.
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	2
Tipo:	MQCFIN

Reason

Descripción:	Código de razón que indica el resultado de la transferencia para este objeto. Sólo se incluye si se proporcionan estructuras MQRR y si el código de razón para MQPUT es MQRC_MULTIPLE_REASONS.
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	2
Tipo:	MQCFIN

MsgId

Descripción:	Identificador del mensaje. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_MSG_ID.
Parámetro PCF:	MQBACF_MSG_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_MSG_ID_LENGTH

CorrelId

Descripción:	Identificador de correlación. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_CORREL_ID.
Parámetro PCF:	MQBACF_CORREL_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_CORREL_ID_LENGTH

GroupId

Descripción:	Identificador de grupo. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_GROUP_ID.
Parámetro PCF:	MQBACF_GROUP_ID

Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_GROUP_ID_LENGTH

Feedback

Descripción: Feedback. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_FEEDBACK.
Parámetro PCF: MQIACF_FEEDBACK
Nivel de rastreo: 2
Tipo: MQCFIN

AccountingToken

Descripción: AccountingToken. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_ACCOUNTING_TOKEN.
Parámetro PCF: MQBACF_ACCOUNTING_TOKEN
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_ACCOUNTING_TOKEN_LENGTH.

MQPUT1

La aplicación ha iniciado la función MQPUT1 de MQI

ObjectType

Descripción: Tipo de objeto transferido en MQOT.ObjectType
Parámetro PCF: MQIACF_OBJECT_TYPE
Nivel de rastreo: 1
Tipo: MQCFIN

ObjectName

Descripción: Nombre del objeto pasado a la llamada MQI antes de que se haya intentado resolver algún nombre de cola.
Parámetro PCF: MQCACF_OBJECT_NAME
Nivel de rastreo: 1
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

ObjectQMgrName

Descripción: Nombre del gestor de colas del objeto transferido a la llamada MQI antes de intentar cualquier resolución de nombre de cola.
Parámetro PCF: MQCACF_OBJECT_Q_MGR_NAME
Nivel de rastreo: 1
Tipo: MQCFST
Longitud: MQ_Q_MGR_NAME_LENGTH

CompCode

Descripción:	Código de terminación que indica el resultado de la operación
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

Reason

Descripción:	Código de razón resultado de la operación
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

PutOptions

Descripción:	Opciones de transferencia de MQPMO.Options
Parámetro PCF:	MQIACF_PUT_OPTIONS
Nivel de rastreo:	1
Tipo:	MQCFIN

AlternateUserId

Descripción:	Sólo se incluye si se especifica MQPMO_ALTERNATE_USER_AUTHORITY.
Parámetro PCF:	MQCACF_ALTERNATE_USERID
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_USER_ID_LENGTH

RecsPresent

Descripción:	Número de registros de nombre de objeto presentes.
Parámetro PCF:	MQIACF_RECS_PRESENT
Nivel de rastreo:	1
Tipo:	MQCFIN

KnownDestCount

Descripción:	Número de colas locales abiertas satisfactoriamente.
Parámetro PCF:	MQIACF_KNOWN_DEST_COUNT
Nivel de rastreo:	1
Tipo:	MQCFIN

UnknownDestCount

Descripción:	Número de colas remotas abiertas satisfactoriamente.
Parámetro PCF:	MQIACF_UNKNOWN_DEST_COUNT
Nivel de rastreo:	1

Tipo: MQCFIN

InvalidDestCount

Descripción: Número de colas que no se han podido abrir.

Parámetro PCF: MQIACF_INVALID_DEST_COUNT

Nivel de rastreo: 1

Tipo: MQCFIN

MsgBuffer

Descripción: Los datos del mensaje.

Parámetro PCF: MQBACF_MESSAGE_DATA

Nivel de rastreo: 1

Tipo: MQCFBS

Longitud: La longitud está gobernada por el parámetro TRACEDATA() definido en la configuración de APPTRACE. Si TRACEDATA=NONE, este parámetro se omite.

MsgLength

Descripción: Longitud del mensaje.

Parámetro PCF: MQIACF_MSG_LENGTH

Nivel de rastreo: 1

Tipo: MQCFIN

HighResTime

Descripción: Tiempo de la operación, en microsegundos, desde la media noche del 1 de enero de 1970 (UTC).

Nota: La precisión de este temporizador variará de acuerdo con el soporte de plataforma para un temporizador de alta resolución.

Parámetro PCF: MQIAMO64_HIGHRES_TIME

Nivel de rastreo: 2

Tipo: MQCFIN64

ResolvedQName

Descripción: Nombre de la cola después de haber realizar la resolución de nombre de cola.

Parámetro PCF: MQCACF_RESOLVED_Q_NAME

Nivel de rastreo: 2

Tipo: MQCFST

Longitud: MQ_Q_NAME_LENGTH

ResolvedQMgrName

Descripción: Nombre del gestor de colas después de haber realizado la resolución de nombre.

Parámetro PCF: MQCACF_RESOLVED_Q_MGR

Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_MGR_NAME_LENGTH

ResolvedLocalQName ⁴

Descripción: Contiene el nombre de cola local después de haber realizado la resolución de nombre.
Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST

ResolvedLocalQMgrName ⁴

Descripción: Contiene el nombre de gestor de colas local después de haber realizado la resolución de nombre.
Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

Descripción: Identificador de seguridad alternativo. Sólo está presente si MQOD.Version es igual o mayor que MOOD_VERSION_3 y MQOD.AlternateSecurityId no es igual a MQSID_NONE.
Parámetro PCF: MQBACF_ALTERNATE_SECURITYID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_SECURITY_ID_LENGTH

ObjectString

Descripción: Nombre de objeto largo. Sólo se incluye si MQOD.Version es igual o mayor que MQOD_VERSION_4 y el campo VSLength de MQOD.ObjectString es MQVS_NULL_TERMINATED o mayor que cero.
Parámetro PCF: MQCACF_OBJECT_STRING
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: La longitud varía.

ResObjectString

Descripción: Nombre de objeto largo después de que el gestor de colas resuelva el nombre proporcionado en el campo ObjectName. Sólo se incluye para los temas y los alias de colas que hacen referencia a un objeto de tema si MQOD.Version es igual o mayor que MQOD_VERSION_4 y VSLength es MQVS_NULL_TERMINATED o mayor que cero.
Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING

Nivel de rastreo: 2
Tipo: MQCFST
Longitud: La longitud varía.

ResolvedType

Descripción: El tipo del objeto (base) resuelto que se está abriendo. Sólo se incluye si MQOD.Version es igual o mayor que MQOD_VERSION_4. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF: MQIACF_RESOLVED_TYPE
Nivel de rastreo: 2
Tipo: MQCFIN

ReportOptions

Descripción: Opciones de informe de mensaje
Parámetro PCF: MQIACF_REPORT
Nivel de rastreo: 2
Tipo: MQCFIN

MsgType

Descripción: Tipo de mensaje
Parámetro PCF: MQIACF_MSG_TYPE
Nivel de rastreo: 2
Tipo: MQCFIN

Expiry

Descripción: Duración del mensaje
Parámetro PCF: MQIACF_EXPIRY
Nivel de rastreo: 2
Tipo: MQCFIN

Format

Descripción: Nombre de formato de datos de mensaje
Parámetro PCF: MQCACH_FORMAT_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_FORMAT_LENGTH

Priority

Descripción: Prioridad de mensaje
Parámetro PCF: MQIACF_PRIORITY
Nivel de rastreo: 2
Tipo: MQCFIN

Persistence

Descripción: Persistencia de los mensajes
Parámetro PCF: MQIACF_PERSISTENCE
Nivel de rastreo: 2
Tipo: MQCFIN

MsgId

Descripción: Identificador de mensaje
Parámetro PCF: MQBACF_MSG_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_MSG_ID_LENGTH

CorrelId

Parámetro PCF: Identificador de correlación
Descripción: MQBACF_CORREL_ID
Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_CORREL_ID_LENGTH

ReplyToQueue

Descripción:
Parámetro PCF: MQCACF_REPLY_TO_Q
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

ReplyToQMgr

Descripción:
Parámetro PCF: MQCACF_REPLY_TO_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQCFST

CodedCharSetId

Descripción: Identificador de juego de caracteres de datos de mensaje
Parámetro PCF: MQIA_CODED_CHAR_SET_ID
Nivel de rastreo: 2
Tipo: MQCFIN

Encoding

Descripción:	Codificación numérica de datos de mensaje.
Parámetro PCF:	MQIACF_ENCODING
Nivel de rastreo:	2
Tipo:	MQCFIN

PutDate

Descripción:	
Parámetro PCF:	MQCACF_PUT_DATE
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_PUT_DATE_LENGTH

PutTime

Descripción:	
Parámetro PCF:	MQCACF_PUT_TIME
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_PUT_TIME_LENGTH

PolicyName

Descripción:	Nombre de la política aplicada al mensaje. Nota: Sólo para mensajes protegidos por AMS.
Parámetro PCF:	MQCA_POLICY_NAME
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_OBJECT_NAME_LENGTH

XmitqMsgId

Descripción:	ID del mensaje en la cabecera de la cola de transmisión. Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.
Parámetro PCF:	MQBACF_XQH_MSG_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_MSG_ID_LENGTH

XmitqCorrelId

Descripción:	ID de correlación del mensaje en la cabecera de cola de transmisión. Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.
Parámetro PCF:	MQBACF_XQH_CORREL_ID

Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_CORREL_ID_LENGTH

XmitqPutTime

Descripción: Hora de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_TIME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_TIME_LENGTH

XmitqPutDate

Descripción: Fecha de transferencia del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_PUT_DATE
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_NAME
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

XmitqRemoteQMgr

Descripción: El destino de cola remota del mensaje en la cabecera de cola de transmisión.
Nota: Sólo cuando Format es MQFMT_XMIT_Q_HEADER.

Parámetro PCF: MQCACF_XQH_REMOTE_Q_MGR
Nivel de rastreo: 2
Tipo: MQCFST
Longitud: MQ_Q_NAME_LENGTH

PutMsgOptsStructure

Descripción: Estructura MQPMO.
Parámetro PCF: MQBACF_MQPMO_STRUCTURE
Nivel de rastreo: 3

Tipo: MQCFBS
Longitud: Longitud en bytes de la estructura MQPMO (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2

Tipo: MQCFIN64

Estructura de cabecera de grupo PCF AppActivityDistList MQPUT1

Si la función MQPUT1 transfiere datos a una lista de distribución, los parámetros de variable incluyen un grupo PCF AppActivityDistList. Para cada una de las colas de la lista de distribución, consulte [“Estructura de cabecera del grupo PCF de la lista de distribución de actividad de aplicación” en la página 259.](#)

El grupo PCF AppActivityDistList combina información de las estructuras MQOR, MQPMR y MQRR para identificar los objetos, y los parámetros PUT indican el resultado de la operación PUT en cada cola. Para operaciones MQPUT1, el grupo AppActivityDistList contiene algunos de los parámetros que se indican a continuación, o todos ellos (CompCode, Reason, ObjectName y ObjectQMgrName están presentes si el código de razón es MQRC_MULTIPLE_REASONS y los demás parámetros se determinan mediante el campo MQPMO.PutMsgRecFields):

CompCode

Descripción: Código de terminación que indica el resultado de la transferencia para este objeto. Sólo se incluye si se proporcionan estructuras MQRR y si el código de razón para MQPUT1 es MQRC_MULTIPLE_REASONS.

Parámetro PCF: MQIACF_COMP_CODE

Nivel de rastreo: 2

Tipo: MQCFIN

Reason

Descripción: Código de razón que indica el resultado de la transferencia para este objeto. Sólo se incluye si se proporcionan estructuras MQRR y si el código de razón para MQPUT1 es MQRC_MULTIPLE_REASONS.

Parámetro PCF: MQIACF_REASON_CODE

Nivel de rastreo: 2

Tipo: MQCFIN

⁴ El parámetro ResolvedLocalQName sólo se incluye si es distinto del parámetro ResolvedQName.

ObjectName

Descripción:	Nombre de una cola en la lista de distribución. Sólo se incluye si se proporcionan estructuras MQOR.
Parámetro PCF:	MQCACF_OBJECT_NAME
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_Q_NAME_LENGTH

MsgId

Descripción:	Identificador del mensaje. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_MSG_ID.
Parámetro PCF:	MQBACF_MSG_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_MSG_ID_LENGTH

CorrelId

Descripción:	Identificador de correlación. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_CORREL_ID.
Parámetro PCF:	MQBACF_CORREL_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_CORREL_ID_LENGTH

GroupId

Descripción:	Identificador de grupo. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_GROUP_ID.
Parámetro PCF:	MQBACF_GROUP_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_GROUP_ID_LENGTH

Feedback

Descripción:	Feedback. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_FEEDBACK.
Parámetro PCF:	MQIACF_FEEDBACK
Nivel de rastreo:	2
Tipo:	MQCFIN

AccountingToken

Descripción:	AccountingToken. Sólo se incluye si se proporcionan estructuras MQPMR y si PutMsgRecFields incluye MQPMRF_ACCOUNTING_TOKEN.
Parámetro PCF:	MQBACF_ACCOUNTING_TOKEN

Nivel de rastreo: 2
Tipo: MQCFBS
Longitud: MQ_ACCOUNTING_TOKEN_LENGTH.

MQSET

La aplicación ha iniciado la función MQSET de MQI

ObjectHandle

Descripción: Manejador de objeto
Parámetro PCF: MQIACF_HOBJ
Nivel de rastreo: 1
Tipo: MQCFIN

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación
Parámetro PCF: MQIACF_REASON_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

SelectorCount

Descripción: La cuenta de selectores que se suministran en la matriz Selectors.
Parámetro PCF: MQIACF_SELECTOR_COUNT
Nivel de rastreo: 2
Tipo: MQCFIN

Selectors

Descripción: Lista de atributos (entero o carácter) cuyos valores se actualizan mediante MQSET.
Parámetro PCF: MQIACF_SELECTORS
Nivel de rastreo: 2
Tipo: MQCFIL

ResolvedQName

Descripción: Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo: 2

Tipo MQCFST
Longitud: MQ_Q_NAME_LENGTH.

ResObjectString

Descripción: Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF: MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo: 2
Tipo MQCFST
Longitud: La longitud varía.

ResolvedType

Descripción: Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF: MQIACF_RESOLVED_TYPE
Nivel de rastreo: 2
Tipo MQCFIN

IntAttrCount

Descripción: Número de atributos enteros que deben actualizarse mediante la operación de definición.
Parámetro PCF: MQIACF_INTATTR_COUNT
Nivel de rastreo: 3
Tipo: MQCFIN

IntAttrrs

Descripción: Valores de atributo entero
Parámetro PCF: MQIACF_INT_ATTRS
Nivel de rastreo: 3
Tipo: MQCFIL
Rango: Este parámetro solo está presente si IntAttrCount es > 0

CharAttrrs

Descripción: Atributos de carácter que deben actualizarse mediante la opción de definición. Los valores se concatenan entre sí.
Parámetro PCF: MQCACF_CHAR_ATTRS
Nivel de rastreo: 3
Tipo: MQCFST
Rango: Este parámetro solo se incluye si CharAttrLength es > 0

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.

Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION

Nivel de rastreo: 2

Tipo: MQCFIN64

MQSUB

La aplicación ha iniciado la función MQSUB de MQI

CompCode

Descripción: Código de terminación que indica el resultado de la operación

Parámetro PCF: MQIACF_COMP_CODE

Nivel de rastreo: 1

Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación

Parámetro PCF: MQIACF_REASON_CODE

Nivel de rastreo: 1

Tipo: MQCFIN

SubHandle

Descripción: Manejador de la suscripción

Parámetro PCF: MQIACF_HSUB

Nivel de rastreo: 1

Tipo: MQCFIN

ObjectHandle

Descripción: Manejador de objeto

Parámetro PCF: MQIACF_HOBJ

Nivel de rastreo: 1

Tipo: MQCFIN

Options

Descripción: Opciones de suscripción

Parámetro PCF: MQIACF_SUB_OPTIONS

Nivel de rastreo: 1

Tipo: MQCFIN

ObjectName

Descripción:	El nombre del objeto.
Parámetro PCF:	MQCACF_OBJECT_NAME
Nivel de rastreo:	1
Tipo:	MQCFST
Longitud:	MQ_Q_NAME_LENGTH

ObjectString

Descripción:	Nombre de objeto largo.
Parámetro PCF:	MQCACF_OBJECT_STRING
Nivel de rastreo:	1
Tipo:	MQCFST
Rango:	Sólo se incluye si el campo VSLength de MQSD.ObjectString es mayor que cero o MQVS_NULL_TERMINATED.
Longitud:	La longitud varía.

AlternateUserId

Descripción:	
Parámetro PCF:	MQCACF_ALTERNATE_USERID
Nivel de rastreo:	2
Tipo:	MQCFST
Rango:	Sólo se incluye si se especifica MQSO_ALTERNATE_USER_AUTHORITY.
Longitud:	MQ_USER_ID_LENGTH

AlternateSecurityId

Descripción:	Identificador de seguridad alternativo.
Parámetro PCF:	MQBACF_ALTERNATE_SECURITYID
Nivel de rastreo:	2
Tipo:	MQCFBS
Rango:	Sólo está presente si se especifica MQSO_ALTERNATE_USER_AUTHORITY y MQSD.AlternateSecurityId no es igual a MQSID_NONE.
Longitud:	MQ_SECURITY_ID_LENGTH

SubName

Descripción:	Nombre de la suscripción
Parámetro PCF:	MQCACF_SUB_NAME
Nivel de rastreo:	2
Tipo:	MQCFST
Rango:	Sólo se incluye si el campo VSLength de MQSD.SubName es mayor que cero o MQVS_NULL_TERMINATED.
Longitud:	La longitud varía.

SubUserData

Descripción:	Datos de usuario de suscripción
Parámetro PCF:	MQCACF_SUB_USER_DATA
Nivel de rastreo:	2
Tipo:	MQCFST
Rango:	Sólo se incluye si el campo VSLength de MQSD.SubName es mayor que cero o MQVS_NULL_TERMINATED.
Longitud:	La longitud varía.

SubCorrelId

Descripción:	Identificador de correlación de suscripción
Parámetro PCF:	MQBACF_SUB_CORREL_ID
Nivel de rastreo:	2
Tipo:	MQCFBS
Longitud:	MQ_CORREL_ID_LENGTH

SelectionString

Descripción:	Serie de selección.
Parámetro PCF:	MQCACF_SELECTION_STRING
Nivel de rastreo:	2
Tipo:	MQCFST
Rango:	Sólo se incluye si el campo VSLength de MQSD. SelectionString es MQVS_NULL_TERMINATED o mayor que cero.
Longitud:	La longitud varía.

ResolvedQName

Descripción:	Nombre de la cola al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_Q.
Parámetro PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	MQ_Q_NAME_LENGTH.

ResObjectString

Descripción:	Nombre del objeto al que hace referencia ObjectHandle, cuando ResolvedType es MQOT_TOPIC.
Parámetro PCF:	MQCACF_RESOLVED_OBJECT_STRING
Nivel de rastreo:	2
Tipo:	MQCFST
Longitud:	La longitud varía.

ResolvedType

Descripción:	Tipo de objeto al que hace referencia ObjectHandle. Los valores posibles son MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parámetro PCF:	MQIACF_RESOLVED_TYPE
Nivel de rastreo:	2
Tipo	MQCFIN

SubDescriptorStructure

Descripción:	Estructura MQSD.
Parámetro PCF:	MQBACF_MQSD_STRUCT
Nivel de rastreo:	3
Tipo:	MQCFBS
Longitud:	Longitud en bytes de la estructura MQSD.

QMGrOpDuration

Descripción:	Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas. La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ. Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF:	MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo:	2
Tipo	MQCFIN64

MQSUBRQ

La aplicación ha iniciado la función MQSUBRQ de MQI

CompCode

Descripción:	Código de terminación que indica el resultado de la operación
Parámetro PCF:	MQIACF_COMP_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

Reason

Descripción:	Código de razón resultado de la operación
Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

SubHandle

Descripción:	Manejador de la suscripción
Parámetro PCF:	MQIACF_HSUB

Nivel de rastreo: 1
Tipo: MQCFIN

SubOptions

Descripción: Opciones de suscripción de MQSB.Options
Parámetro PCF: MQIACF_SUBRQ_OPTIONS
Nivel de rastreo: 2
Tipo: MQCFIN

Action

Descripción: Acción de solicitud de suscripción (MQSR_*)
Parámetro PCF: MQIACF_SUBRQ_ACTION
Nivel de rastreo: 2
Tipo: MQCFIN

NumPubs

Descripción: Número de publicaciones enviadas como resultado de esta llamada (de MQSB.NumPubs)
Parámetro PCF: MQIACF_NUM_PUBS
Nivel de rastreo: 2
Tipo: MQCFIN

QMGrOpDuration

Descripción: Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas.

La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ.
Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.

Parámetro PCF: MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo: 2
Tipo: MQCFIN64

MQSTAT

La aplicación ha iniciado la función MQSTAT de MQI

CompCode

Descripción: Código de terminación que indica el resultado de la operación
Parámetro PCF: MQIACF_COMP_CODE
Nivel de rastreo: 1
Tipo: MQCFIN

Reason

Descripción: Código de razón resultado de la operación

Parámetro PCF:	MQIACF_REASON_CODE
Nivel de rastreo:	1
Tipo:	MQCFIN

Type

Descripción:	Tipo de información de estado que se solicita
Parámetro PCF:	MQIACF_STATUS_TYPE
Nivel de rastreo:	2
Tipo:	MQCFIN

StatusStructure

Descripción:	Estructura de MQSTS
Parámetro PCF:	MQBACF_MQSTS_STRUCT
Nivel de rastreo:	3
Tipo:	MQCFBS
Longitud:	Longitud en bytes de la estructura MQSTS (el tamaño real depende de la versión de la estructura)

QMGrOpDuration

Descripción:	Duración aproximada de la llamada de API, en microsegundos, dentro del gestor de colas. La duración no incluye el tiempo pasado fuera del gestor de colas. Por ejemplo, el tiempo empleado como cliente IBM MQ. Nota: La exactitud de este temporizador varía según la plataforma que utilice su empresa.
Parámetro PCF:	MQIAMO64_QMGR_OP_DURATION
Nivel de rastreo:	2
Tipo:	MQCFIN64

Parámetros de variable para operaciones XA de actividad de la aplicación

Las operaciones XA son llamadas de API que las aplicaciones pueden realizar para habilitar MQ para que participe en una transición. Los parámetros para cada operación se definen en la siguiente sección.

El nivel de rastreo indica el nivel de granularidad de rastreo que se necesita para los parámetros que se incluirán en el rastreo. Los valores de nivel de rastreo posibles son:

1. Bajo

El parámetro se incluye cuando se configura el rastreo de actividad "low", "medium" o "high" para una aplicación. Este valor significa que siempre se incluye un parámetro en el grupo `AppActivityData` de la operación. Este conjunto de parámetros es suficiente para rastrear las llamadas MQI que una aplicación realiza y para ver si son satisfactorios.

2. Medio

El parámetro sólo se incluye en el grupo `AppActivityData` para la operación cuando se configura el rastreo de actividad "medium" o "high" para una aplicación. Este conjunto de parámetros añade información sobre los recursos, por ejemplo, los nombres de colas y temas utilizados por la aplicación.

3. Alto

El parámetro sólo se incluye en el grupo `AppActivityData` para la operación cuando se configura el rastreo de actividad "high" para una aplicación. Este conjunto de parámetros incluye volcados de memoria de las estructuras que se pasan a las funciones de MQI y XA. Por este motivo, contiene más información sobre los parámetros utilizados en llamadas MQI y XA. Los valores de memoria de la estructura son copias superficiales de las estructuras. Para evitar intentos erróneos de anulación de referencia de los punteros, los valores de puntero en las estructuras se establecen en NULL.

Nota: La versión de la estructura que se vuelca no es necesariamente idéntica a la versión utilizada por una aplicación. La estructura se puede modificar mediante una salida cruzada de API, mediante el código de rastreo de actividad o mediante el gestor de colas. Un gestor de colas puede modificar una estructura en una versión posterior, pero el gestor de colas nunca la cambia a una versión anterior de la estructura. En caso de hacerlo, se corre el riesgo de perder datos.

AXREG

La aplicación ha iniciado la función AXREG AX

XID

Descripción:	Estructura XID
Parámetro PCF:	MQBACF_XA_XID
Nivel de rastreo:	1
Tipo:	MQCFBS
Longitud:	Sizeof(XID)

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE
Nivel de rastreo:	1
Tipo:	MQCFIN

AXUNREG

La aplicación ha iniciado la función AXUNREG AX

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1

Tipo: MQCFIN

Flags

Descripción: Flags
Parámetro PCF: MQIACF_XA_FLAGS
Nivel de rastreo: 1
Tipo: MQCFIN

XARetCode

Descripción: Código de retorno
Parámetro PCF: MQIACF_XA_RETCODE
Nivel de rastreo: 1
Tipo: MQCFIN

XACLOSE

La aplicación ha iniciado la función XACLOSE AX

Xa_info

Descripción: Información utilizada para inicializar el gestor de recursos.
Parámetro PCF: MQCACF_XA_INFO
Nivel de rastreo: 1
Tipo: MQCFST

Rmid

Descripción: Identificador del gestor de recursos
Parámetro PCF: MQIACF_XA_RMID
Nivel de rastreo: 1
Tipo: MQCFIN

Flags

Descripción: Flags
Parámetro PCF: MQIACF_XA_FLAGS
Nivel de rastreo: 1
Tipo: MQCFIN

XARetCode

Descripción: Código de retorno
Parámetro PCF: MQIACF_XA_RETCODE
Nivel de rastreo: 1
Tipo: MQCFIN

XACOMMIT

La aplicación ha iniciado la función XACOMMIT AX

XID

Descripción:	Estructura XID
Parámetro PCF:	MQBACF_XA_XID
Nivel de rastreo:	1
Tipo:	MQCFBS
Longitud:	Sizeof(XID)

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE
Nivel de rastreo:	1
Tipo:	MQCFIN

XACOMPLETE

La aplicación ha iniciado la función XACOMPLETE AX

Handle

Descripción:	Manejador para la operación asíncrona
Parámetro PCF:	MQIACF_XA_HANDLE
Nivel de rastreo:	1
Tipo:	MQCFIN

Retval

Descripción:	Valor de retorno de la función asíncrona
Parámetro PCF:	MQIACF_XA_RETVAL
Nivel de rastreo:	1
Tipo:	MQCFINMQCFBS

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID

Nivel de rastreo: 1
Tipo: MQCFIN

Flags

Descripción: Flags
Parámetro PCF: MQIACF_XA_FLAGS
Nivel de rastreo: 1
Tipo: MQCFIN

XARetCode

Descripción: Código de retorno
Parámetro PCF: MQIACF_XA_RETCODE
Nivel de rastreo: 1
Tipo: MQCFIN

XAEND

La aplicación ha iniciado la función XAEND AX

XID

Descripción: Estructura XID
Parámetro PCF: MQBACF_XA_XID
Nivel de rastreo: 1
Tipo: MQCFBS
Longitud: Sizeof(XID)

Rmid

Descripción: Identificador del gestor de recursos
Parámetro PCF: MQIACF_XA_RMID
Nivel de rastreo: 1
Tipo: MQCFIN

Flags

Descripción: Flags
Parámetro PCF: MQIACF_XA_FLAGS
Nivel de rastreo: 1
Tipo: MQCFIN

XARetCode

Descripción: Código de retorno
Parámetro PCF: MQIACF_XA_RETCODE
Nivel de rastreo: 1
Tipo: MQCFIN

XAFORGET

La aplicación ha iniciado la función AXREG AX

XID

Descripción:	Estructura XID
Parámetro PCF:	MQBACF_XA_XID
Nivel de rastreo:	1
Tipo:	MQCFBS
Longitud:	Sizeof(XID)

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE
Nivel de rastreo:	1
Tipo:	MQCFIN

XAOPEN

La aplicación ha iniciado la función XAOPEN AX

Xa_info

Descripción:	Información utilizada para inicializar el gestor de recursos.
Parámetro PCF:	MQCACF_XA_INFO
Nivel de rastreo:	1
Tipo:	MQCFST

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE
Nivel de rastreo:	1
Tipo:	MQCFIN

XAPREPARE

La aplicación ha iniciado la función XAPREPARE AX

XID

Descripción:	Estructura XID
Parámetro PCF:	MQBACF_XA_XID
Nivel de rastreo:	1
Tipo:	MQCFBS
Longitud:	Sizeof(XID)

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE
Nivel de rastreo:	1
Tipo:	MQCFIN

XARECOVER

La aplicación ha iniciado la función XARECOVER AX

Count

Descripción: Recuento de XID
Parámetro PCF: MQIACF_XA_COUNT
Nivel de rastreo: 1
Tipo: MQCFIN

XIDs

Descripción: Estructuras XID

Nota: Existen varias instancias de este parámetro PCF, una para cada estructura XID y hasta el recuento de XID.

Parámetro PCF: MQBACF_XA_XID
Nivel de rastreo: 1
Tipo: MQCFBS
Longitud: Sizeof(XID)

Rmid

Descripción: Identificador del gestor de recursos
Parámetro PCF: MQIACF_XA_RMID
Nivel de rastreo: 1
Tipo: MQCFIN

Flags

Descripción: Flags
Parámetro PCF: MQIACF_XA_FLAGS
Nivel de rastreo: 1
Tipo: MQCFIN

XARetCode

Descripción: Código de retorno
Parámetro PCF: MQIACF_XA_RETCODE
Nivel de rastreo: 1
Tipo: MQCFIN

XAROLLBACK

La aplicación ha iniciado la función XAROLLBACK AX

XID

Descripción: Estructura XID
Parámetro PCF: MQBACF_XA_XID
Nivel de rastreo: 1
Tipo: MQCFBS
Longitud: Sizeof(XID)

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE
Nivel de rastreo:	1
Tipo:	MQCFIN

XASTART

La aplicación ha iniciado la función XASTART AX

XID

Descripción:	Estructura XID
Parámetro PCF:	MQBACF_XA_XID
Nivel de rastreo:	1
Tipo:	MQCFBS
Longitud:	Sizeof(XID)

Rmid

Descripción:	Identificador del gestor de recursos
Parámetro PCF:	MQIACF_XA_RMID
Nivel de rastreo:	1
Tipo:	MQCFIN

Flags

Descripción:	Flags
Parámetro PCF:	MQIACF_XA_FLAGS
Nivel de rastreo:	1
Tipo:	MQCFIN

XARetCode

Descripción:	Código de retorno
Parámetro PCF:	MQIACF_XA_RETCODE

Nivel de rastreo: 1
Tipo: MQCFIN

Multi Temas del sistema para la supervisión y el rastreo de actividades

Los temas del sistema en los árboles de temas del gestor de colas se utilizan para la supervisión de recursos (algunos de los cuales son similares al contenido de los mensajes de estadísticas) y como una forma de consumir el rastreo de actividad de la aplicación.

Rama \$SYS/MQ del árbol de temas del gestor de colas

Cada árbol de temas del gestor de colas contiene la rama \$SYS/MQ . El gestor de colas publica en series de tema en esta rama. Un usuario autorizado puede suscribirse a estas series de tema para recibir información sobre el gestor de colas y la actividad de este. Estos temas del sistema se utilizan para la supervisión y el rastreo de la actividad de la aplicación. Para obtener más información, consulte [Árboles de temas](#).

La raíz de la rama \$SYS/MQ se representa mediante el objeto de tema SYSTEM.ADMIN.TOPIC. La rama \$SYS/MQ del árbol de temas está aislada del resto del árbol de temas de las maneras siguientes:

- Una suscripción realizada con caracteres comodín en un punto más alto del árbol que \$SYS/MQ no coincide con ninguna serie de temas dentro de la rama \$SYS/MQ. La operación de comodín para SYSTEM.ADMIN.TOPIC está establecida en "Bloquear" y no se puede modificar. Esta limitación también se aplica al utilizar caracteres comodín con el mandato de **runmqsc** DISPLAY TPSTATUS para mostrar nodos en el árbol de temas. Para ver nodos de temas dentro de la rama \$SYS/MQ, inicie la serie de tema con \$SYS/MQ. Por ejemplo, utilice \$SYS/MQ/# para ver todos los nodos.
- Debe tener autorización en \$SYS/MQ o más allá para que se le otorgue la autorización para utilizar el árbol de temas de \$SYS/MQ. La autorización para suscribirse a una serie de tema se basa en la autorización que se otorga para un objeto de tema administrado en la serie de tema del árbol de temas o en un punto superior. Las autorizaciones que se otorgan en la misma raíz (SYSTEM.BASE.TOPIC) otorgarían a un usuario autorización para todas las series de tema. No obstante, en el caso de la rama \$SYS/MQ, el acceso otorgado a un punto más alto que \$SYS/MQ no se aplica a las series de tema de \$SYS/MQ.
- La rama \$SYS/MQ del árbol de temas está aislada de los atributos de tema establecidos en un punto más alto del árbol. SYSTEM.ADMIN.TOPIC no hereda ningún atributo de un objeto de tema definido más arriba en el árbol de temas. Por ejemplo, el cambio de los atributos de SYSTEM.BASE.TOPIC no afecta al comportamiento de la rama \$SYS/MQ.

Todas las series de tema que comienzan con \$SYS/MQ se reservan para uso de IBM MQ. Estas series de tema tienen las restricciones siguientes:

- No puede habilitar la multidifusión a partir de la rama \$SYS/MQ del árbol de temas.
- No se admite la agrupación en clúster para la rama \$SYS/MQ.
- No se puede establecer el mecanismo de suscripción de proxy en "forzar".
- Las aplicaciones no pueden publicar en una serie de tema \$SYS/MQ.
- El ámbito de publicación y suscripción solo se establece de forma predeterminada para el gestor de colas local.
- No se pueden utilizar caracteres comodín en los puntos siguientes:
 - \$SYS/MQ/
 - \$SYS/MQ/INFO
 - \$SYS/MQ/INFO/QMGR
 - \$SYS/MQ/INFO/QMGR/*nombreGC*
 - \$SYS/MQ/INFO/QMGR/*nombreGC*/ActivityTrace

Los intentos de utilizar caracteres comodín en estos puntos provocarán un error de suscripción con razón MQRC_ADMIN_TOPIC_STRING_ERROR.

API de supervisión de rendimiento disponible a través de publicación/suscripción

Puede supervisar las estadísticas del gestor de colas utilizando el mecanismo de publicación/suscripción simple y la aplicación de ejemplo amqsrua o su propia aplicación. Las estadísticas se publican en un tema del sistema bajo \$SYS/MQ/INFO/QMGR que ayuda a los usuarios a supervisar recursos. Estas estadísticas se pueden visualizar ejecutando la aplicación de ejemplo amqsrua, o escribiendo una aplicación que se suscriba al tema del sistema de supervisión de recursos de una forma similar a **amqsrua**.

Conceptos relacionados

“Rastreo de actividad de la aplicación” en la página 207

El rastreo de actividad de la aplicación genera información detallada sobre el comportamiento de las aplicaciones conectadas a un gestor de colas. Rastrea el comportamiento de una aplicación y proporciona una vista detallada de los parámetros utilizados por una aplicación a medida que interactúa con los recursos de IBM MQ. También muestra la secuencia de llamadas MQI emitidas por una aplicación.

Supervisión del uso de recursos del sistema mediante el mandato amqsrua

Puede utilizar el mandato **amqsrua** para consultar datos de rendimiento que están relacionados con el uso de recursos del sistema de un gestor de colas.

Acerca de esta tarea

La aplicación de ejemplo amqsrua muestra una forma de consumir las aplicaciones de supervisión de IBM MQ y mostrar los datos de rendimiento que publican los gestores de colas. Estos datos pueden incluir información sobre el uso de CPU, memoria y disco. También puede ver datos equivalentes a los datos de estadísticas de PCF STATMQI. Los datos se publican cada 10 segundos y se notifican mientras se ejecuta el mandato.

Puede ejecutar el mandato con sólo el nombre del gestor de colas e interactivamente paso a paso para elegir los parámetros **CLASS**, luego **TYPE** y luego **object** que están disponibles para el gestor de colas en cada paso. Si conoce los nombres de **CLASS**, **TYPE** y **object** para los que desea ver información, puede especificarlos cuando ejecute el mandato **amqsrua**.

De forma predeterminada, la aplicación amqsrua busca estadísticas publicadas por el gestor de colas bajo el árbol de temas \$SYS/MQ/INFO/QMGR. Otros componentes o aplicaciones pueden utilizar un mecanismo similar de publicación con un punto de partida de tema diferente. Desde IBM MQ 9.1.0, puede utilizar el parámetro **-p** para especificar dónde desea que amqsrua busque las estadísticas para estos otros componentes en ambos, Linux y Windows.

-m

Nombre del gestor de colas. El gestor de colas debe estar en ejecución. Si no especifica un nombre de gestor de colas, se muestran las estadísticas para el gestor de colas predeterminado.

-c

El nombre de CLASS. Las publicaciones de uso de recursos de IBM MQ están asociadas a una clase. Las clases representan el nivel superior en el árbol de metadatos que describe la información de uso de recursos disponible.

CPU

Devuelve información sobre el uso de CPU.

DISK

Devuelve información sobre el uso de disco.

STATMQI

Devuelve información sobre el uso de MQI.

STATQ

Devuelve información sobre el uso de MQI por cola.

STATAPP

Devuelve información sobre el uso de estadística para la aplicación especificada.

Consulte [“Desarrollo de su propia aplicación de supervisión de recursos”](#) en la página 298 para obtener información sobre cómo especificar el árbol de temas para los metadatos, y [Using the application name in supported programming languages](#) para los caracteres que puede utilizar en los nombres de aplicaciones.

Nota: Utilice la opción **-o** para especificar el nombre de aplicación que se está supervisando para STATAPP y el nombre de cola para STATQ.

-t

El nombre de TYPE. Las publicaciones de uso de recursos de IBM MQ están asociadas a un tipo dentro de una clase. Cada publicación incluye la clase y el tipo que permite que se encuentren las definiciones de `class/type/element` y las publicaciones resultantes que se van a manejar. Las descripciones de `class/type/element` se publican como metadatos al iniciarse el gestor de colas.

-o

El nombre del objeto. Las publicaciones de uso de recursos son mensajes PCF que constan de una secuencia de elementos PCF. Los elementos PCF que se publican para cada par de clase/tipo se anuncian en los metadatos. Se almacena una hoja en el árbol que describe cada elemento, permitiendo así que se procesen los elementos.

-p

El prefijo de metadatos. Especifique un punto de inicio de árbol de temas donde **amqsrua** puede buscar estadísticas publicadas por gestores de colas. El árbol de temas predeterminado es `$SYS/MQ/INFO/QMGR`, pero otros componentes o aplicaciones pueden publicar estadísticas bajo un punto de partida de árbol de temas diferente.

-n

El recuento de publicaciones. Puede especificar cuántos informes se devuelven antes de que finalice el mandato. Los datos se publican aproximadamente cada diez segundos, así si especifica un valor de 50, el mandato devuelve 50 informes en 500 segundos. Si no especifica este parámetro, el mandato se ejecuta hasta que se produce un error o hasta que concluya el gestor de colas.

-s

Cola modelo. Opcionalmente, especifique una cola de modelo que utilizar. (De forma predeterminada, **amqsrua** utiliza `SYSTEM.DEFAULT.MODEL.QUEUE`.)

-h

Utilización

Procedimiento

1. Desde el directorio de ejemplo, emita el mandato siguiente para mostrar los datos disponibles para el gestor de colas:

-  En Linux, `MQ_INSTALLATION_PATH/samp/bin`:

```
./amqsrua -m QMgrName
```

-  En Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`:

```
amqsrua -m QMgrName
```

donde *Nombre_gestor_colas* especifica el nombre del gestor colas que desea consultar. El gestor de colas debe estar en ejecución. Si no especifica un nombre de gestor de colas, se utiliza el gestor de colas predeterminado.

Están disponibles las opciones siguientes:

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
```

```
Enter Class selection
==>
```

2. Desde la lista de opciones CLASS, especifique STATMQI.

```
==> STATMQI
CONNDISC : MQCONN and MQDISC
OPENCLOSE : MQOPEN and MQCLOSE
INQSET : MQINQ and MQSET
PUT : MQPUT
GET : MQGET
SYNCPOINT : Commit and rollback
SUBSCRIBE : Subscribe
PUBLISH : Publish
Enter Type selection
==>
```

3. Desde la lista de opciones TYPE, especifique PUT.

```
==>PUT
Publication received PutDate:20170329 PutTime:17045485 Interval:4 minutes,13.978 seconds
Interval total MQPUT/MQPUT1 count 22
Interval total MQPUT/MQPUT1 byte count 25284 100/sec
Non-persistent message MQPUT count 22
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 25284 100/sec
Put persistent messages - byte count 0
MQSTAT count 0

Publication received PutDate:20170329 PutTime:17050485 Interval:10.001 seconds
Interval total MQPUT/MQPUT1 count 1
Interval total MQPUT/MQPUT1 byte count 524 52/sec
Non-persistent message MQPUT count 1
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 524 52/sec
Put persistent messages - byte count 0
MQSTAT count 0
```

Resultados

Ha utilizado la aplicación de ejemplo **amqsrxa** de forma interactiva para consultar las estadísticas que publican los gestores de colas en los temas del sistema bajo el prefijo de metadatos \$SYS/MQ/INFO/QMGR.

Nota: Las clases y los tipos de publicaciones de recurso que están disponibles para gestores de colas podrían ser diferentes en función de su configuración, versión y plataforma. Utilice **amqsrxa** de forma interactiva para encontrar las clases, tipos y elementos que están disponibles para el gestor de colas específico.

Qué hacer a continuación

Para desarrollar su propia aplicación de supervisión, consulte [“Desarrollo de su propia aplicación de supervisión de recursos”](#) en la página 298.

Tareas relacionadas

[Preparación y ejecución de programas de ejemplo en Windows](#)

[Preparación y ejecución de programas de ejemplo en UNIX y Linux](#)

Desarrollo de su propia aplicación de supervisión de recursos

Puede desarrollar su propia aplicación para supervisar recursos del sistema.

Cada gestor de colas publica los datos de uso de recursos en temas. Estos datos los consumen los suscriptores a dichos temas. Cuando se inicia un gestor de colas, el gestor de colas publica un conjunto de mensajes en metatemas. Estos mensajes describen qué temas de uso de recursos admite el gestor de colas, así como el contenido de los mensajes publicados en dichos temas. Las herramientas administrativas pueden suscribirse a los metadatos para descubrir qué información de uso de recursos está disponible, y sobre qué temas y, después, suscribirse a los temas anunciados.

El árbol de temas para los metadatos tiene la estructura siguiente:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

Utilice la estructura siguiente para los metadatos del árbol de temas:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

para clarificar que está especificando el recurso que se va a supervisar, que tiene el máximo de 28 caracteres para el nombre de aplicación y para evitar confusión con los atributos internos de tipo de metadatos.

Si desea una lista de posibles clases, consulte [“Supervisión del uso de recursos del sistema mediante el mandato amqsrua”](#) en la página 296.

El código fuente para la aplicación amqsrua se proporciona como un ejemplo de IBM MQ. Puede utilizar esta aplicación de ejemplo como guía para crear su propia aplicación de supervisión. Puede recuperar el código fuente del ejemplo desde una instalación de cliente de IBM MQ. El archivo de origen se denomina amqsruaa.c y se encuentra en el directorio de ejemplos:

- ▶ **Linux** ▶ **AIX** En plataformas AIX and Linux, `MQ_INSTALLATION_PATH/samp/`
- ▶ **Windows** En plataformas Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\`

La aplicación amqsrua se suscribe a temas de uso de recursos de IBM MQ y da formato a los datos PCF publicados resultantes. El origen de la aplicación proporciona un ejemplo básico de cómo suscribirse a y consumir este tipo de datos administrativos. La aplicación amqsrua completa las tareas siguientes:

- Crea una suscripción no duradera a los temas identificados por los parámetros de entrada.
- Llama a MQGET repetidamente para obtener mensajes de los temas y escribe en stdout.
- Escribe un mensaje por cada razón de MQI (distinta de MQRC_NONE).
- Se detiene si hay un código de terminación de MQI de MQCC_FAILED, o cuando se ha consumido el número solicitado de publicaciones de uso de recursos.

▶ **Multi** Métricas publicadas en los temas del sistema

Las métricas se clasifican en clases y se subcategorizan en tipos. Hay varias métricas publicadas en cada clase de métrica y tipo.

Índice

- [“CPU \(unidades de proceso central de la plataforma\)”](#) en la página 300
- [“DISK \(almacenes de datos persistentes de plataforma\)”](#) en la página 301
- [“STATMQI \(estadísticas de uso de API\)”](#) en la página 302
- [“STATQ \(estadísticas de uso por cola de API\)”](#) en la página 304
- [“STATAPP \(estadísticas de uso por aplicación\)”](#) en la página 305
- [“NHAREPLICA \(estadísticas de HA nativa por instancia\)”](#) en la página 305

Consulte [“Supervisión del uso de recursos del sistema mediante el mandato amqsrua”](#) en la [página 296](#) para obtener información sobre cómo recopilar datos para las opciones listadas, con la excepción de NHAREPLICA.

También puede utilizar el [mandato ALTER QMGR](#) para supervisar STATMQI y STATQ a nivel de gestor de colas, o el atributo de cola local STATQ para supervisar colas individuales; consulte [ALTER QUEUE](#) para esta opción.

CPU (unidades de proceso central de la plataforma)

Introducción

Donde las estadísticas hacen referencia al *intervalo actual*, este es el intervalo definido en el mensaje publicado por el parámetro MQIAMO64_MONITOR_INTERVAL .

Las estadísticas se publican normalmente cada 10 segundos, el *intervalo publicado*, siempre que haya al menos un suscriptor activo, pero el intervalo preciso debe tomarse siempre del mensaje.

Importante: A menos que se especifique lo contrario, las métricas son valores absolutos para el punto en el tiempo cuando se capturan.

SystemSummary (rendimiento de CPU - toda la plataforma)

Porcentaje de tiempo de CPU de usuario X%

Porcentaje promedio de tiempo (empleado durante el último intervalo de 10 segundos) utilizado por la CPU cuando estaba en código no privilegiado.

Porcentaje de tiempo de CPU del sistema X%

Porcentaje promedio de tiempo (utilizado durante el último intervalo de 10 segundos) utilizado por la CPU cuando estaba en código privilegiado.

Carga de CPU - promedio de un minuto X

El promedio de carga de un minuto. "Promedio de carga" es un término de toda la industria, pero el valor exacto reportado puede diferir entre plataformas.

Carga de CPU - promedio de cinco minutos X

El promedio de carga de cinco minutos. "Promedio de carga" es un término de toda la industria, pero el valor exacto reportado puede diferir entre plataformas.

Carga de CPU - promedio de quince minutos X

El promedio de carga de quince minutos. "Promedio de carga" es un término de toda la industria, pero el valor exacto reportado puede diferir entre plataformas.

Resumen del sistema de CPU

Porcentaje libre de RAM X%

Total de bytes de RAM XMB

Porcentaje libre de RAM X%

Total de bytes de RAM XMB

QMGrSummary (rendimiento de CPU - gestor de en ejecución)

Tiempo de CPU de usuario - estimación de porcentaje para el gestor de colas X%

Porcentaje promedio de tiempo (empleado durante el último intervalo de 10 segundos) utilizado por la CPU cuando los procesos de este gestor de colas estaban en código no privilegiado.

Tiempo de CPU del sistema - estimación de porcentaje para el gestor de colas X%

Porcentaje promedio de tiempo (empleado durante el último intervalo de 10 segundos) utilizado por la CPU cuando los procesos de este gestor de colas estaban en código privilegiado.

Total de bytes de RAM - estimación para el gestor de colas XMB

Se trata de una aproximación de la memoria utilizada por el gestor de colas.

DISK (almacenes de datos persistentes de plataforma)

SystemSummary y *QMGrSummary* son valores absolutos en el momento de la captura. Consulte la [Introducción](#) para obtener detalles del *intervalo publicado*.

SystemSummary (uso de disco - toda la plataforma)

Sistema de archivos de errores MQ - bytes en uso XMB

Sistema de archivos de errores MQ - espacio libre X%

Recuento de archivos FDC de MQ X

Sistema de archivos de rastreo MQ - bytes en uso XMB

Sistema de archivos de rastreo de MQ . espacio libre X%

QMGrSummary (uso de disco - gestores de colas en ejecución)

Sistema de archivos del gestor de colas - bytes en uso XMB

Sistema de archivos del gestor de colas - espacio libre X%

Registro (uso de disco - registro de recuperación del gestor de colas)

Registro - bytes en uso X

Registro - bytes máx. X

El número máximo de bytes que se pueden escribir en el registro si todas las extensiones primarias y secundarias estaban llenas. Es menor que el tamaño del sistema de archivos de registro

Sistema de archivos de registro - bytes en uso X

Sistema de archivos de registro - bytes máx. X

Bytes físicos de registro grabados para el intervalo actual X.

Consulte la [Introducción](#) para ver la definición de *intervalo actual*.

Registro-bytes lógicos grabados para el *intervalo actual* X

Registro - latencia de escritura X uSeg

Promedio escalonado que representa el tiempo que tarda una sola grabación en disco.

Donde **LogWriteIntegrity**=*TripleWrite*, el número físico de bytes grabados en disco es mayor que los bytes lógicos grabados.

Tamaño de grabación de registro X, también promedio continuo.

Registro-ocupado por extensiones en espera de ser archivadas X.

Solo se publica cuando **logtype**= *linear* y **LogManagement** = *archive*. Consulte [Stanza de registro del archivo qm.ini](#) para obtener más información.

Espacio de registro en MB necesario para la recuperación de soporte X.

Solo se publica cuando **logtype**= *linear*.

Espacio de registro en MB ocupado por extensiones reutilizables X

Sólo se publica cuando **logtype**= *linear* y **LogManagement** = *automatic*. Consulte [Stanza de registro del archivo qm.ini](#) para obtener más información.

Registro-espacio primario actual en uso X%.

Espacio de archivos de registro en uso como porcentaje de registros primarios. Este valor puede ser superior al 100%.

Registro-utilización de espacio primario de carga de trabajo X%.

El porcentaje de espacio de archivos de registro en uso como promedio continuo sobre el historial reciente.

STATMQI (estadísticas de uso de API)

Todas las estadísticas de uso de API reflejan apariciones y/o porcentajes para el intervalo publicado. Consulte la [Introducción](#) para ver la definición de *intervalo publicado*.

Las estadísticas generan un recuento del número de llamadas MQI fallidas, pero no todas las llamadas MQI fallidas aparecen en estas estadísticas; de hecho, las anomalías de no todas las llamadas MQI tienen sus estadísticas registradas. Esto se debe a que muchas razones por las que las llamadas MQI fallan se diagnostican antes de que la llamada MQI alcance los valores internos del gestor de colas donde se registran las estadísticas.

Un ejemplo de esto es MQRC_HCONN_ERROR devuelto a una aplicación cliente. Si una aplicación cliente pasa un **hconn** erróneo, el cliente MQ diagnostica ese error y devuelve MQRC_HCONN_ERROR sin pasar la llamada MQI al gestor de colas. Por lo tanto, la llamada MQI fallida nunca aparece en las estadísticas registradas por el gestor de colas.

Las estadísticas de llamadas MQI fallidas son útiles porque permiten a los clientes resolver problemas de aplicaciones mal escritas que generan llamadas MQI fallidas innecesarias, lo que afecta al rendimiento. Algunos ejemplos de razones de error para varias llamadas MQI que se registran en las estadísticas:

- MQCONN/MQCONN/MQOPEN devuelve 2035 MQRC_NOT_AUTHORIZED cuando lo diagnostica el gestor de colas, no el cliente. Por ejemplo, ejecutando **amqspu**t como nadie.
- MQPUT/MQPUT1 devuelve 2053 MQRC_Q_FULL porque se ha sobrepasado MAXDEPTH.
- MQGET devuelve 2033 MQRC_NO_MSG_AVAILABLE al examinar o obtener de forma destructiva de una cola vacía
- MQSUBRQ devuelve 2437 MQRC_NO_RETAINED_MSG porque no hay ningún mensaje retenido

CONNDISC (MQCONN y MQDISC)

Recuento de MQCONN/MQCONN X

Recuento de MQCONN/MQCONN anómalo X

Conexiones simultánea - marca de límite superior X

Recuento de MQDISC X

OPENCLOSE (MQOPEN y MQCLOSE)

Recuento de MQOPEN X Y/seg.

Recuento de MQOPEN anómalo X

Recuento de MQCLOSE X Y/seg.

Recuento de MQCLOSE anómalo X

INQSET (MQINQ y MQSET)

Recuento de MQINQ X

Recuento de MQINQ anómalo X

Recuento de MQSET X

Recuento de MQSET anómalo X

PUT (MQPUT)

Recuento total de MQPUT/MQPUT1 de intervalo X

Recuento total de bytes MQPUT/MQPUT1 de intervalo X Y/seg.

Recuento de MQPUT de mensaje no persistente X

Recuento de MQPUT de mensaje persistente X

Recuento de MQPUT anómalo X

Recuento de MQPUT1 de mensajes no persistentes X

Recuento de MQPUT1 de mensajes persistentes X

Recuento de MQPUT1 anómalo X

Entrega de mensajes no persistentes - recuento de bytes X Y/seg.

Entrega de mensajes persistentes - recuento de bytes *X*

Recuento de MQSTAT *X*

GET (MQGET)

Total de obtenciones destructivas de intervalo - recuento *X*

Recuento total de bytes de obtenciones destructivas de intervalo *X Y/seg.*

Recuento de obtenciones destructivas de mensajes no persistentes *X*

Obtenciones destructivas de mensajes persistentes - recuento *X*

Recuento de errores de MQGET *X*

Obtención de mensajes no persistentes - recuento de bytes *X Y/seg.*

Obtención de mensajes persistentes - recuento de bytes *X*

Examen de mensajes no persistentes - recuento *X*

Examen de mensajes persistentes - recuento *X*

Recuento de exámenes anómalos *X*

Examen de mensajes no persistentes - recuento de bytes *X Y/seg.*

Examen de mensajes persistentes - recuento de bytes *X*

Recuento de mensajes caducados *X*

Recuento de colas depuradas *X*

Recuento de MQCB *X*

Recuento de MQCB fallidos *X*

Recuento de MQCTL *X*

SYNCPOINT (confirmar y retrotraer)

Recuento de confirmaciones *X*

Recuento de retrotracciones *X*

SUBSCRIBE (suscribirse)

Crear recuento de suscripciones duraderas *X*

Recuento de modificaciones de suscripciones duraderas *X*

Recuento de reanudaciones de suscripciones duraderas *X*

Recuento de creaciones de suscripciones no duraderas *X*

Recuento de errores de creación/modificación/reanudación de suscripciones *X*

Recuento de supresiones de suscripciones duraderas *X*

Recuento de supresiones de suscripciones no duraderas *X*

Recuento de errores de supresión de suscripciones *X*

Recuento de MQSUBRQ *X*

Recuento de MQSUBRQ fallidos *X*

Suscriptor duradero - marca de límite superior *X*

Suscriptor duradero - marca de límite inferior *X*

Suscriptor no duradero - marca de límite superior *X*

Suscriptor no duradero - marca de límite inferior *X*

PUBLISH (publicar)

Total de MQPUT/MQPUT1 de tema de intervalo *X*

Total de bytes entregados de tema de intervalo *X Y/seg.*

Publicado en suscriptores - recuento de mensajes *X*

Publicado en suscriptores - recuento de bytes *X*

No persistentes - recuento de MQPUT/MQPUT1 de tema *X*

Persistentes - recuento de MQPUT/MQPUT1 de tema X

Recuento de MQPUT/MQPUT1 de tema fallidos X

STATQ (estadísticas de uso por cola de API)

GENERAL (General)

mensajes caducados X (movido de GET para IBM MQ 9.3.0 y versiones posteriores de CD)

recuento de depuraciones de cola X (movido de GET para IBM MQ 9.3.0 y versiones posteriores de CD)

promedio de tiempo en cola X uSeg (movido de GET para IBM MQ 9.3.0 y versiones posteriores de CD)

Profundidad de cola X (movido de GET para IBM MQ 9.3.0 y versiones posteriores de CD)

OPENCLOSE (MQOPEN y MQCLOSE)

Recuento de MQOPEN X

Recuento de MQCLOSE X

INQSET (MQINQ y MQSET)

Recuento de MQINQ X

Recuento de MQSET X

PUT (MQPUT y MQPUT1)

Recuento de MQPUT/MQPUT1 X

Recuento de bytes de MQPUT X

Recuento de mensajes no persistentes de MQPUT X

Recuento de mensajes persistentes de MQPUT X

recuento de MQPUT retrotraídos X

Recuento de mensajes no persistentes de MQPUT1 X

Recuento de mensajes persistentes de MQPUT1 X

recuento de bytes no persistentes X

recuento de bytes persistentes X

contención de bloqueo X%

Porcentaje de intentos de bloquear la cola que han resultado en la espera de que otro proceso libere primero el bloqueo. Es probable que disminuir la contención de bloqueo aumente el rendimiento máximo del sistema porque tomar un bloqueo que no está bloqueado actualmente es más eficaz que esperar a que se libere un bloqueo.

entregas en cola evitadas X%

Si un mensaje se coloca en una cola cuando hay un método de obtención en espera, es posible que el mensaje no sea necesario ponerlo en cola, ya que puede ser posible que se pase al método de obtención inmediatamente. Por lo tanto, se dice que este mensaje ha evitado la cola, y "colocación en cola evitada" es el recuento de dichos mensajes. Es probable que el aumento de la evitación de colas aumente el rendimiento máximo del sistema, ya que evita el coste de colocar el mensaje en la cola y volver a bajarlo.

bytes en cola evitados X%

Si un mensaje se coloca en una cola cuando hay un método de obtención en espera, es posible que el mensaje no sea necesario ponerlo en cola, ya que puede ser posible que se pase al método de obtención inmediatamente. Por lo tanto, se dice que este mensaje ha evitado la cola, y "bytes evitados de cola" es el recuento de dichos bytes. Es probable que el aumento de la evitación de colas aumente el rendimiento máximo del sistema, ya que evita el coste de colocar el mensaje en la cola y volver a bajarlo.

GET (MQGET)

Recuento de MQGET X

Recuento de bytes de MQGET X
Recuento de mensajes no persistentes de MQGET destructivo X
Recuento de mensajes persistentes de MQGET destructivo X
recuento de MQGET retrotraídos X
Recuento de bytes no persistentes de MQGET destructivo X
Recuento de bytes persistentes de MQGET destructivo X
Recuento de exámenes de mensajes no persistentes de MQGET X
Recuento de exámenes de mensajes persistentes de MQGET X
Recuento de bytes no persistentes de examen de MQGET X
Recuento de bytes persistentes de examen de MQGET X
mensajes caducados X (se ha movido a GENERAL desde IBM MQ 9.3)
recuento de cola depurada X (movido a GENERAL desde IBM MQ 9.3)
promedio de tiempo en cola X uSec (movido a GENERAL desde IBM MQ 9.3)
Profundidad de cola X (se ha movido a GENERAL desde IBM MQ 9.3)
Errores de MQGET destructivo X
Error de MQGET destructivo con MQRC_NO_MSG_AVAILABLE X
Error de MQGET destructivo con MQRC_TRUNCATED_MSG_FAILED X
Error de examen de MQGET X
Error de examen de MQGET con MQRC_NO_MSG_AVAILABLE X
Error de examen de MQGET con MQRC_TRUNCATED_MSG_FAILED X





STATAPP (estadísticas de uso por aplicación)

INSTANCE (estadísticas de instancia)

Recuento de instancias X absoluto
Recuento de instancias móviles X absoluto
Recuento de déficit de instancia X absoluto
Instancias iniciadas X intervalo
La instancia de salida iniciada mueve X intervalo
La instancia de salida completada mueve X intervalo
Instancias finalizadas durante la reconexión X intervalo
Instancias finalizadas X intervalo

NHAREPLICA (estadísticas de HA nativa por instancia)

REPLICATION (estadísticas de réplica)

Bytes de anotaciones cronológicas síncronas enviados X
Bytes de registro de captura enviados X
 Bytes de registro comprimidos síncronos enviados X
 Bytes de registro comprimidos de captura enviados X
 Bytes de registro no comprimidos síncronos enviados X
 Bytes de registro no comprimidos de captura enviados X
Latencia de promedio de acuse de recibo de grabación de registro X uSec
Tamaño promedio de acuse de recibo de grabación de registro X
Bytes pendientes X

Información relacionada

Métricas publicadas cuando se utiliza IBM MQ Operator

Supervisión en tiempo real

La supervisión en tiempo real es una técnica que permite determinar el estado actual de las colas y los canales en un gestor de colas. La información devuelta es precisa en el momento en el que se ha emitido el mandato.

Hay varios mandatos disponibles que, cuando se emiten, devuelven información en tiempo real acerca de las colas y los canales. Puede devolverse información para una o más colas o canales y la cantidad puede variar. La supervisión en tiempo real puede utilizarse en las siguientes tareas:

- Ayudar a los administradores del sistema a comprender el estado estable de su sistema IBM MQ. Esta ayuda con el diagnóstico de problemas si se produce un problema en el sistema.
- Determinar la condición del gestor de colas en cualquier momento, aun cuando no se haya detectado ningún problema o suceso específico.
- Ayudar a determinar la causa de un problema en el sistema.

Con la supervisión de real-time, se puede devolver información colas o canales. La cantidad de información en tiempo real que se devuelve está controlada por los atributos del gestor de colas, de la cola y del canal.

- Para supervisar una cola, debe emitir mandatos para garantizar que la cola recibe servicio correctamente. Antes de utilizar algunos de los atributos de cola, debe habilitarlos para la supervisión en tiempo real.
- Puede supervisar un canal emitiendo mandatos para garantizar que el canal se ejecuta correctamente. Para poder utilizar algunos de los atributos de canal, debe habilitarlos para la supervisión en tiempo real.

La supervisión en tiempo real para colas y canales es adicional a la supervisión del rendimiento y de los sucesos de canal, e independiente de ésta.

Atributos que controlan la supervisión en tiempo real

Algunos atributos de estado de cola y canal contienen información de supervisión, si se habilita la supervisión en tiempo real. Si no se habilita la supervisión en tiempo real, estos atributos de supervisión no contienen ninguna información de supervisión. En los ejemplos se demuestra cómo se pueden utilizar estos atributos de estado de cola y de canal.

Puede habilitar o inhabilitar la supervisión en tiempo real para colas o canales individuales, o para varios canales o colas. Para controlar colas o canales individuales, establezca el atributo de cola MONQ o el atributo de canal MONCHL, para habilitar o inhabilitar la supervisión en tiempo real. Para controlar varios canales o colas juntos, habilite o inhabilite la supervisión en tiempo real del nivel de gestor de colas, mediante los atributos de gestor de colas MONQ y MONCHL. Para todos los objetos de cola y canal que tengan un atributo de supervisión que se haya especificado con el valor predeterminado, QMGR, la supervisión en tiempo real se controla en el nivel de gestor de colas.

Los canales de clúster emisor definidos automáticamente no son objetos de IBM MQ, por lo que no tienen atributos como los objetos de canal. Para controlar los canales de clúster emisor definidos automáticamente, utilice el atributo de gestor de colas MONACLS. Este atributo determina si los canales de clúster emisor que se definen automáticamente en un gestor de colas están habilitados o inhabilitados para la supervisión de canales.

Para la supervisión de canales en tiempo real, puede establecer el atributo MONCHL en uno de los tres niveles de supervisión siguientes: baja, intermedia o alta. Puede establecer el nivel de supervisión a nivel de objeto, o bien a nivel de gestor de colas. La elección del nivel depende del sistema. Recopilar datos de supervisión puede requerir ciertas instrucciones que resultan relativamente caras a nivel computacional como, por ejemplo, obtener la hora del sistema. Para poder reducir el efecto de la supervisión en tiempo

real, las opciones de supervisión baja e intermedia miden una muestra de los datos a intervalos regulares, en lugar de recopilar datos todo el tiempo. En la [Tabla 30](#) en la [página 307](#) se resumen los niveles de supervisión disponibles para la supervisión de canales en tiempo real:

Nivel	Descripción	Utilización
Bajo	Medir una pequeña muestra de los datos, a intervalos regulares.	Para los objetos que procesan un alto volumen de mensajes.
Medio	Medir una muestra de los datos, a intervalos regulares.	Para la mayoría de los objetos.
Alto	Medir todos los datos, a intervalos regulares.	Para los objetos que procesan sólo unos pocos mensajes por segundo, en que la información más reciente es importante.

Para la supervisión de colas en tiempo real, puede establecer el atributo MONQ en uno de los tres niveles de supervisión siguientes, baja, intermedia o alta. No obstante, no hay ninguna distinción entre estos valores. Todos los valores habilitan la recopilación de datos, pero no afectan al tamaño de la muestra.

Ejemplos

En los ejemplos siguientes se demuestra cómo establecer la cola, el canal, y los atributos de gestor de colas necesarios para poder controlar el nivel de supervisión. Para todos los ejemplos, cuando la supervisión está habilitada, los objetos de cola y de canal tienen un nivel supervisión intermedia.

1. Para habilitar la supervisión tanto de colas como canales, para todas las colas y todos los canales del nivel de gestor de colas, utilice los mandatos siguientes:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

2. Para habilitar la supervisión para todas las colas y todos los canales, con la excepción de la cola local, Q1, y el canal emisor, QM1 . TO . QM2, utilice los mandatos siguientes:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

3. Para inhabilitar la supervisión para todas las colas y todos los canales, con la excepción de la cola local, Q1, y el canal emisor, QM1 . TO . QM2, utilice los mandatos siguientes:

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

4. Para inhabilitar la supervisión tanto de colas como canales, para todas las colas y todos los canales, independientemente de los atributos de objeto individuales, utilice el mandato siguiente:

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

5. Para controlar las posibilidades de supervisión de los canales de clúster emisor definidos automáticamente, utilice el mandato siguiente:

```
ALTER QMGR MONACLS(MEDIUM)
```

6. Para especificar que los canales de clúster emisor definidos automáticamente utilizarán el valor del gestor de colas para la supervisión de canales, utilice el mandato siguiente:

```
ALTER QMGR MONACLS(QMGR)
```

Conceptos relacionados

[“Supervisión en tiempo real” en la página 306](#)

La supervisión en tiempo real es una técnica que permite determinar el estado actual de las colas y los canales en un gestor de colas. La información devuelta es precisa en el momento en el que se ha emitido el mandato.

[“Using IBM MQ online monitoring” en la página 327](#)

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

Tareas relacionadas

[“Visualización de datos de supervisión de cola y canal” en la página 308](#)

Para visualizar información de supervisión en tiempo real para una cola o un canal, utilice IBM MQ Explorer o el mandato MQSC apropiado. En algunos campos de supervisión se muestran pares de valores de indicador separados por comas, que le ayudan a supervisar el funcionamiento del gestor de colas. En los ejemplos se muestra cómo se pueden visualizar los datos de supervisión.

[Visualización y modificación de atributos del gestor de colas](#)

[Supervisión \(MONCHL\)](#)

Visualización de datos de supervisión de cola y canal

Para visualizar información de supervisión en tiempo real para una cola o un canal, utilice IBM MQ Explorer o el mandato MQSC apropiado. En algunos campos de supervisión se muestran pares de valores de indicador separados por comas, que le ayudan a supervisar el funcionamiento del gestor de colas. En los ejemplos se muestra cómo se pueden visualizar los datos de supervisión.

Acerca de esta tarea

Los campos de supervisión en los que se muestra un par de valores separados por coma proporcionan indicadores a corto y largo plazo para el tiempo medido desde que se habilitó la supervisión para el objeto, o desde el momento en que se inició el gestor de colas:

- El indicador de corto plazo es el primer valor del par, y se calcula de forma que a las medidas más recientes se les otorga una ponderación superior y tendrán un impacto mayor en este valor. Esto proporciona una indicación de la tendencia reciente de las medidas tomadas.
- El indicador de largo plazo es el segundo valor del par, y se calcula de forma que a las mediciones más recientes no se les otorgan una ponderación tan alta. Esto proporciona una indicación de la actividad a largo plazo sobre el rendimiento de un recurso.

Estos valores de indicador resultan muy útiles para detectar cambios en el funcionamiento del gestor de colas. Esto requiere conocer los tiempos que muestran estos indicadores cuando se usan de una forma normal, para poder detectar los aumentos en dichos tiempos. Al recopilar y comprobar dichos valores regularmente, podrá detectar fluctuaciones en el funcionamiento del gestor de colas. Esto puede indicar que se ha producido un cambio en el rendimiento.

Obtenga la información en tiempo real tal como se indica a continuación:

Procedimiento

1. Para visualizar la información de supervisión en tiempo real de una cola, utilice IBM MQ Explorer o el mandato MQSC DISPLAY QSTATUS, especificando el parámetro opcional MONITOR.
2. Para visualizar información de supervisión en tiempo real para un canal, utilice IBM MQ Explorer o el mandato MQSC DISPLAY CHSTATUS, especificando el parámetro opcional MONITOR.

Ejemplo

La cola, Q1, tiene el atributo MONQ establecido en el valor predeterminado, QMGR, y el gestor de colas que posee la cola tiene el atributo MONQ establecido en MEDIUM. Para visualizar los campos de supervisión recopilados para esta cola, utilice el mandato siguiente:

```
DISPLAY QSTATUS(Q1) MONITOR
```

Los campos de supervisión y el nivel de supervisión de la cola, Q1, se muestran tal como se indica a continuación:

```
QSTATUS(Q1)
TYPE(Queue)
MONQ(MEDIUM)
QTIME(11892157,24052785)
MSGAGE(37)
LPUTDATE(2005-03-02)
LPUTTIME(09.52.13)
LGETDATE(2005-03-02)
LGETTIME(09.51.02)
```

El canal emisor, QM1.T0.QM2, tiene el atributo MONCHL establecido en el valor predeterminado, QMGR, y el gestor de colas que posee la cola tiene el atributo MONCHL establecido en MEDIUM. Para visualizar los campos de supervisión recopilados para este canal emisor, utilice el mandato siguiente:

```
DISPLAY CHSTATUS(QM1.T0.QM2) MONITOR
```

Los campos de supervisión y el nivel de supervisión del canal emisor, QM1.T0.QM2, se muestran tal como se indica a continuación:

```
CHSTATUS(QM1.T0.QM2)
XMITQ(Q1)
CONNAM(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)
```

Conceptos relacionados

[“Supervisión en tiempo real” en la página 306](#)

La supervisión en tiempo real es una técnica que permite determinar el estado actual de las colas y los canales en un gestor de colas. La información devuelta es precisa en el momento en el que se ha emitido el mandato.

Referencia relacionada

[DISPLAYQSTATUS](#)

Supervisión de colas

Utilice esta página para ver las tareas que le ayudarán a resolver un problema que haya surgido en una cola y la aplicación que le presta servicios. Hay disponibles varias opciones de supervisión para poder determinar el problema.

Con frecuencia, el primer signo de que existe un problema en una cola a la que se ofrecen servicios, es que el número de mensajes de la cola (CURDEPTH) aumenta. Si espera que haya un incremento en ciertos momentos del día, o bajo determinadas cargas de trabajo, es posible que un aumento en el número de

mensajes no indique se exista un problema. No obstante, si no tiene explicación para el aumento del número de mensajes, es posible que desee investigar el motivo.

Puede tener una cola de aplicación en la que exista un problema con la aplicación, o una cola de transmisión en la que haya un problema con el canal. Las opciones de supervisión adicionales están disponibles cuando la aplicación que presta servicios a la cola es un canal.

En los ejemplos siguientes se investigan los problemas surgidos con una cola determinada, denominada Q1, y se describen los campos que se pueden observar en la salida de varios mandatos:

Determinación de si la aplicación tiene la cola abierta

Si tiene un problema con una cola, compruebe si la aplicación tiene la cola abierta.

Acerca de esta tarea

Realice los pasos siguientes para determinar si la aplicación tiene la cola abierta:

Procedimiento

1. Asegúrese de que la aplicación que se ejecute contra la cola sea la aplicación que espera. Emita el mandato siguiente para la cola en cuestión:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

En la salida, busque el campo APPLTAG y compruebe que aparezca el nombre de la aplicación. Si el nombre de la aplicación no aparece, o si no hay ninguna salida, inicie la aplicación.

2. Si la cola es una cola de transmisión, busque en la salida el campo CHANNEL.
Si el nombre del canal no aparece en el campo CHANNEL, determine si se está ejecutando el canal.
3. Asegúrese de que la aplicación que se ejecute contra la cola tenga la cola abierta poder recibir entradas. Emita el mandato siguiente:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

En la salida, busque el campo IPPROCS para ver si alguna aplicación tiene la cola abierta para poder recibir entradas. Si el valor es 0, y se trata de una cola de aplicación de usuario, asegúrese de que la aplicación abra la cola para las entradas, y así poder obtener los mensajes que provengan de fuera de la cola.

Comprobar que los mensajes de la cola están disponibles

Si tiene un gran número de mensajes en la cola, y la aplicación no está procesando ninguno de dichos mensajes, compruebe si los mensajes de la cola están disponibles para la aplicación.

Acerca de esta tarea

Efectúe los pasos siguientes para investigar por qué la aplicación no procesa los mensajes de la cola:

Procedimiento

1. Asegúrese de que su aplicación no pida un ID de mensaje, o de correlación, específico cuando debería estar procesando todos los mensajes de la cola.
2. Aunque la profundidad actual de la cola pueda mostrar que en la cola hay un número creciente de mensajes, es posible que algunos mensajes de la cola no estén disponibles para que una aplicación los obtenga, porque no están confirmados; la profundidad actual incluye el número de MQPUT no confirmadas de mensajes en la cola. Emita el mandato siguiente:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

En la salida, busque el campo UNCOM para ver si hay algún mensaje sin confirmar en la cola.

3. Si la aplicación intenta obtener algún mensaje de la cola, compruebe si la aplicación que efectúa la transferencia está confirmando los mensajes correctamente. Emita el siguiente mandato para averiguar los nombres de las aplicaciones que están transfiriendo mensajes a esta cola:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) OPENTYPE(OUTPUT)
```

4. A continuación, emita el mandato siguiente, insertándolo en *appltag*, el valor APPLTAG de la salida del mandato anterior:

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

Esto muestra cuándo se inició la unidad de trabajo, y le ayudará a descubrir si la aplicación está creando una unidad de trabajo de larga ejecución. Si la aplicación que efectúa la transferencia es un canal, es posible que desee investigar por qué un proceso por lotes tarda mucho tiempo en completarse.

Comprobar si la aplicación está obteniendo mensajes de la cola

Si surge un problema con una cola y la aplicación que da servicio a dicha cola, compruebe si la aplicación está obteniendo mensajes de la cola.

Acerca de esta tarea

Para comprobar si la aplicación está obteniendo mensajes de la cola, efectúe las comprobaciones siguientes:

Procedimiento

1. Asegúrese de que la aplicación que se está ejecutando para la cola está realmente procesando mensajes de la cola. Emita el mandato siguiente:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

En la salida, observe los campos de LGETDATE y LGETTIME, en los que se muestra cuándo se efectuó la última operación de tipo get de la cola.

2. Si la última operación get de esta cola tuvo lugar hace más tiempo de lo esperado, asegúrese de que la aplicación esté procesando los mensajes correctamente.

Si la aplicación es un canal, compruebe si los mensajes se mueven a través del mismo.

Determinar si la aplicación puede procesar mensajes lo suficientemente rápido

Si los mensajes se crean en la cola, pero las otras comprobaciones no han encontrado ningún problema de proceso, compruebe si la aplicación puede procesar los mensajes con la rapidez suficiente. Si la aplicación es un canal, compruebe que el canal pueda procesar los mensajes con la rapidez suficiente.

Acerca de esta tarea

Para determinar si la aplicación procesa los mensajes lo suficientemente rápido, efectúe las pruebas siguientes:

Procedimiento

1. Emita el mandato siguiente periódicamente para recopilar datos de rendimiento sobre el canal:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Si los valores de los indicadores QTIME son altos, o aumentan durante el período, y ya ha descartado la posibilidad de que se trate de unidades de trabajo de larga ejecución, comprobando que los

mensajes de la cola estén disponibles, es posible que la aplicación que ejecuta la operación get no esté al tanto de las aplicaciones que ejecuten la operación put.

2. Si la aplicación que ejecuta la operación get no puede estar al tanto de las aplicaciones que ejecutan la operación put, puede añadir otra aplicación que ejecute la operación get para procesar la cola.

Si se puede añadir otra aplicación que ejecute la operación get depende del diseño de la aplicación, y de si la cola puede compartirse entre más de una aplicación. Las características como, por ejemplo, la agrupación de mensajes o la obtención mediante el ID de correlación pueden ayudar a garantizar que dos aplicaciones puedan procesar una cola simultáneamente.

Comprobación de la cola cuando la profundidad actual no aumenta

Aunque la profundidad actual de la cola de no aumente, aún puede resultar útil supervisar la cola para comprobar si la aplicación procesa los mensajes correctamente.

Acerca de esta tarea

Para recopilar datos de rendimiento sobre la cola: emita, periódicamente, el mandato siguiente:

Procedimiento

Emita el mandato siguiente periódicamente:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

En la salida, si el valor en MSGAGE aumenta durante el período de tiempo, y la aplicación se ha diseñado para procesar todos los mensajes, esto podría indicar que algunos mensajes no se estén procesando en absoluto.

Supervisión de canales

Utilice esta página para ver las tareas que le ayudarán a resolver un problema que haya surgido en una cola de transmisión y el canal que le presta servicios. Hay disponibles varias opciones de supervisión de canales para poder determinar el problema.

Con frecuencia, el primer signo de que existe un problema en una cola a la que se ofrecen servicios, es que el número de mensajes de la cola (CURDEPTH) aumenta. Si espera que haya un incremento en ciertos momentos del día, o bajo determinadas cargas de trabajo, es posible que un aumento en el número de mensajes no indique se exista un problema. No obstante, si no tiene explicación para el aumento del número de mensajes, es posible que desee investigar el motivo.

Es posible que exista un problema con el canal que preste servicios a una cola de transmisión. Hay disponibles varias opciones de supervisión de canales que le pueden ayudar a poder determinar el problema.

En los ejemplos siguientes se investigan los problemas surgidos con una cola de transmisión denominada QM2 y un canal llamado QM1.TO.QM2. Este canal se utiliza para enviar mensajes desde el gestor de colas, QM1, al gestor de colas, QM2. La definición de canal del gestor de colas QM1 es un canal emisor o servidor, y la definición de canal del gestor de colas QM2 es un canal receptor o peticionario.

Determinación de si se está ejecutando el canal

Si tiene un problema con una cola de transmisión, compruebe si el canal se está ejecutando.

Acerca de esta tarea

Efectúe los pasos siguientes para comprobar el estado del canal que esté dando servicio a la cola de transmisión:

Procedimiento

1. Emita el mandato siguiente para averiguar qué canal se espera que procese la cola de transmisión QM2:


```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```

En este ejemplo, en la salida de este mandato se muestra que el canal que da servicio a la cola de transmisión es QM1.TO.QM2.

2. Emita el mandato siguiente para determinar el estado del canal, QM1.TO.QM2:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

3. Inspeccione el campo STATUS de la salida del mandato **CHSTATUS**:

- Si el valor del campo STATUS es RUNNING, compruebe que el canal esté moviendo los mensajes.
- Si en la salida del mandato no se muestra ningún estado o el valor del campo STATUS es STOPPED, RETRY, BINDING o REQUESTING, efectúe el paso pertinente, tal como se indica a continuación:

4. Opcional: Si en el valor del campo STATUS no muestra ningún estado, significa el canal está inactivo, por lo que deberá efectuar los pasos siguientes:

- a) Si un desencadenante debe haber iniciado automáticamente el canal, compruebe que los mensajes de la cola de transmisión estén disponibles.

Si hay mensajes disponibles en la cola de transmisión, compruebe que los valores de desencadenante de la cola de transmisión sean los correctos.

- b) Emita el mandato siguiente para volver a iniciar el canal manualmente:

```
START CHANNEL(QM1.TO.QM2)
```


5. Opcional: Si el valor del campo STATUS es STOPPED, efectúe los pasos siguientes:

- a) Compruebe los registros de errores para determinar por qué se ha detenido el canal. Si el canal se detenido debido a un error, corrija el problema.

Asegúrese también de que el canal tenga valores especificados para los atributos de reintento: *SHORTRTY* y *LONGRTY*. En el caso de producirse anomalías transitorias como, por ejemplo, errores de red, el canal tratará de reiniciarse automáticamente.

- b) Emita el mandato siguiente para volver a iniciar el canal manualmente:

```
START CHANNEL(QM1.TO.QM2)
```

 En IBM MQ for z/OS, puede detectar cuándo un usuario detiene un canal utilizando mensajes de suceso de mandato.

6. Opcional: Si el valor del campo STATUS es RETRY, efectúe los pasos siguientes:

- a) Compruebe los registros de errores para poder identificar el error y, a continuación, corrija el problema.

- b) Emita el mandato siguiente para volver a iniciar el canal manualmente:

```
START CHANNEL(QM1.TO.QM2)
```

o bien espere a que el canal se conecte de forma satisfactoria en su próximo reintento.

7. Opcional: Si el valor del campo STATUS es BINDING o REQUESTING, significa que el canal aún no se ha conectado de forma satisfactoria al socio. Efectúe los pasos siguientes:

- a) Emita el mandato siguiente, en ambos extremos del canal, para determinar el subestado del canal:


```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

Nota:

- i) En algunos casos, puede haber un subestado sólo en un extremo del canal.

ii) Muchos subestados son transitorios, por lo que debe emitir el mandato algunas veces para poder detectar si un canal está atascado en un subestado determinado.

b) Consulte [Tabla 31](#) en la [página 314](#) para determinar qué acción debe llevar a cabo:

<i>Tabla 31. Subestados vistos con el estado BINDING o REQUESTING</i>		
Iniciando un subestado de MCA ¹	Respondiendo al subestado de MCA ²	Notas
NAMESERVER		El MCA iniciador está a la espera de que finalice una solicitud de servidor de nombres. Asegúrese de que se ha especificado el nombre de host correcto en el atributo de canal, CONNAME y de que los servidores de nombres se hayan configurado correctamente.
SCYEXIT	SCYEXIT	Los MCA se hallan actualmente en un estado de tipo <i>en conversación</i> , a través de una salida de seguridad. Para obtener más información, consulte “Determinación de si el canal puede procesar mensajes con la rapidez suficiente” en la página 316 .
	CHADEXIT	Actualmente se está ejecutando la salida de definición automática de canal. Para obtener más información, consulte “Determinación de si el canal puede procesar mensajes con la rapidez suficiente” en la página 316 .
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	Las salidas se llaman durante el inicio del canal para MQXR_INIT. Revise el proceso de esta parte de la salida, si tarda mucho tiempo. Para obtener más información, consulte “Determinación de si el canal puede procesar mensajes con la rapidez suficiente” en la página 316 .
SERIALIZE	SERIALIZE	Este subestado sólo se aplica a canales que tengan una disposición de SHARED.
NETCONNECT		Este subestado se muestra si hay un retraso en la conexión debido a una configuración de red incorrecta.
SSLHANDSHAKE	SSLHANDSHAKE	Un reconocimiento TLS consta de una serie de envíos y recepciones. Si los tiempos de red son lentos, o la conexión para buscar las CRL es lenta, esto afecta al tiempo que se tarda en hacer el reconocimiento.  En IBM MQ for z/OS, este subestado también puede indicar un valor insuficiente de SSLTASKS.

Notas:

- i) El MCA iniciador es el extremo del canal que ha iniciado la conversación. Puede tratarse de remitentes, clústeres emisores, peticionarios y servidores totalmente calificados. En un par de servidor-peticionario, es el extremo desde el que se inició el canal.
- ii) El MCA respondedor es el extremo del canal que ha respondido a la solicitud de inicio de la conversación. Puede tratarse de receptores de clústeres receptores, peticionarios (cuando se inicia el servidor o emisor), servidores (cuando se inicia el peticionario) y remitentes (en un par de canales de peticionario-remitente y devolución de llamada).

Comprobación de que el canal mueve mensajes

Si tiene un problema con una cola de transmisión, compruebe que el canal mueve mensajes.

Antes de empezar

Emita el mandato `DIS CHSTATUS(QM1.TO.QM2) ALL`. Si el valor del campo `STATUS` es `RUNNING`, significa que el canal se ha conectado de forma satisfactoria al sistema socio.

Compruebe que no haya mensajes sin confirmar en la cola de transmisión, tal como se describe en [“Comprobar que los mensajes de la cola están disponibles” en la página 310](#).

Acerca de esta tarea

Si hay mensajes disponibles que el canal puede obtener y enviar, realice las comprobaciones siguientes:

Procedimiento

1. En la salida del mandato de visualizar estado del canal, `DIS CHSTATUS(QM1.TO.QM2) ALL`, observe los campos siguientes:

MSGS

Número de mensajes enviados o recibidos (o, para canales de conexión con el servidor, número de llamadas MQI gestionadas) durante la sesión (desde que se inició el canal).

BUFSENT

Número de almacenamientos intermedios de transmisión que se han enviado. Incluye las transmisiones para enviar sólo información de control.

BYTSENT

Número de bytes enviados durante esta sesión (desde que se inició el canal). Incluye información de control enviada por el agente de canal de mensajes.

LSTMSGDA

Fecha en que se envió el último mensaje o se gestionó la última llamada MQI. Vea `LSTMSGTI`.

LSTMSGTI

Hora en que se envió el último mensaje o se gestionó la última llamada MQI. Para un emisor o servidor, es la hora en que se envió el último mensaje (o el último segmento del mismo, si estaba segmentado). Para un peticionario o receptor, es la hora en que se transfirió el último mensaje a la cola de destino. Para un canal de conexión de servidor, es la hora en que terminó la última llamada MQI.

CURMSGGS

Para un canal emisor, es el número de mensajes que se ha enviado en el lote actual. Para un canal receptor, es el número de mensajes que se han recibido en el lote actual. Cuando el lote se confirma, el valor se restablece en cero tanto en el canal emisor como en el receptor.

Estado

El estado del canal, que puede ser `Starting`, `Binding`, `Initializing`, `Running`, `Stopping`, `Retrying`, `Paused`, `Stopped` o `Requesting`.

SUBSTATE

La acción que está realizando el canal en este momento.

INDOUBT

Indica si el canal está actualmente pendiente o no. Esto es sólo `YES` mientras el agente de canal de mensajes remitente está esperando un acuse de recibo de que un lote de mensajes que ha enviado se ha recibido correctamente. El valor es `NO` en todas las demás ocasiones, incluido el periodo durante el cual se envían los mensajes, pero antes de haber solicitado un acuse de recibo. Para un canal receptor, el valor siempre es `NO`.

2. Determine si el canal ha enviado algún mensaje desde que se inició. Si ha enviado algún mensaje, determine cuándo se envió el último.
3. El canal puede haber iniciado un proceso por lotes que aún no se ha completado, tal como se indica mediante un valor distinto de cero en `CURMSGGS`. Si `INDOUBT` es `YES`, el canal está a la espera de recibir acuse de recibo que indica que el otro extremo del canal ha recibido el proceso por lotes. Observe el campo `SUBSTATE` de la salida, y consulte [Tabla 32 en la página 316](#):

Tabla 32. Subestados MCA de emisor y receptor

SUBSTATE de emisor	SUBSTATE de receptor	Notas
MQGET	RECEIVE	Estados normales de un canal en reposo.
SEND	RECEIVE	Normalmente, SEND es un estado transitorio. Si se visualiza SEND, significa que se han llenado los almacenamientos intermedios del protocolo de comunicaciones. Esto puede indicar que existe un problema de red.
RECEIVE		Si se visualiza el emisor en el subestado RECEIVE para cualquier período de tiempo, está a la espera de una respuesta, ya sea para una finalización de proceso por lotes o una pulsación. Es posible que desee comprobar por qué un proceso por lotes tarda tanto tiempo en completarse.

Nota: También es posible que desee determinar si el canal puede procesar mensajes lo suficientemente rápido, especialmente si el canal tiene un subestado asociado al proceso de salida.

Comprobar porque un lote tarda mucho tiempo en completarse

Las razones por las cuales un lote puede tardar mucho tiempo en completarse son, entre otras, una red lenta o un canal que está utilizando el proceso de reintento de mensajes.

Acerca de esta tarea

Cuando un canal emisor ha enviado un lote de mensajes, espera a que el receptor confirme dicho lote, a menos que el canal se haya canalizado. Los factores descritos en esta tarea pueden afectar a la cantidad de tiempo que espera el canal emisor.

Procedimiento

- Compruebe si la red es lenta.

El valor de NETTIME es la cantidad de tiempo, indicada en microsegundos, que se tarda en enviar una solicitud de fin de lote al extremo remoto del canal y recibir una respuesta, menos el tiempo necesario para procesar la solicitud de fin de lote. Este valor puede ser grande por cualquiera de las siguientes razones:

 - La red es lenta. Una red lenta puede afectar al tiempo que se tarda en completar un lote. Las mediciones que aparecen como resultado en los indicadores del campo NETTIME se miden al final de un lote. No obstante, el primer lote que se vea afectado por una disminución de la velocidad de la red no aparecerá indicado con un cambio en el valor NETTIME, porque se mide al final del lote.
 - Las solicitudes se ponen en cola en el extremo remoto, por ejemplo, un canal puede reintentar una transferencia o una solicitud de transferencia puede ser lenta debido a la E/S del conjunto de páginas. Una vez que se han completado las solicitudes en cola, se mide la duración del final de la solicitud por lotes. Por lo tanto, si obtiene un valor NETTIME grande, compruebe si hay un proceso inusual en el extremo remoto.
- Compruebe si el canal utiliza el reintento de mensajes.

Si el canal receptor no puede poner un mensaje en una cola de destino, puede utilizar el proceso de reintento de mensajes, en lugar de colocar inmediatamente el mensaje en una cola de mensajes no entregados. El proceso de reintento puede provocar que se ralentice el lote. Entre los intentos de MQPUT, el canal tendrá el valor STATUS(PAUSED), que indica que está esperando a que transcurra el intervalo de reintento de mensajes.

Determinación de si el canal puede procesar mensajes con la rapidez suficiente

Si los mensajes se crean en la cola de transmisión, pero no ha encontrado ningún problema de proceso, determine si el canal puede procesar los mensajes con la suficiente rapidez.

Antes de empezar

Emita el mandato siguiente repetidamente durante un período de tiempo para recopilar datos de rendimiento sobre el canal:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

Acerca de esta tarea

Confirme que no haya mensajes sin confirmar en la cola de transmisión, tal como se describe en [“Comprobar que los mensajes de la cola están disponibles”](#) en la [página 310](#) y, a continuación, compruebe el campo XQTIME de la salida del mandato de visualizar estado del canal. Cuando los valores de los indicadores XQTIME sean altos de forma coherente, o aumenten a lo largo del período de medición, la indicación es que el canal no sigue el ritmo de las aplicaciones que llevan a cabo la operación de tipo put.

Realice las pruebas siguientes:

Procedimiento

1. Compruebe si se están procesando las salidas.

Si se utilizan las salidas en el canal que entrega estos mensajes, se pueden añadir al tiempo empleado en el proceso de los mensajes. Para identificar si éste es el caso, realice las verificaciones siguientes:

a) En la salida del mandato `DIS CHSTATUS(QM1.TO.QM2) ALL`, compruebe el campo EXITTIME.

Si el tiempo empleado en las salidas es mayor que el esperado, revise el proceso de las salidas para comprobar si existen bucles o cualquier proceso adicional innecesarios, especialmente en las salidas de mensajes, de envío y de recepción. Este tipo de proceso afecta a todos los mensajes que se desplazan a través del canal.

b) En la salida del mandato `DIS CHSTATUS(QM1.TO.QM2) ALL`, compruebe el campo SUBSTATE.

Si el canal tiene uno de los subestados siguientes durante un tiempo significativo, revise el proceso de las salidas:

- SCYEXIT
- RCVEXIT
- SENDEXIT
- MSGEXIT
- MREXIT

Para obtener más información sobre subestados de canal, consulte la tabla [Tabla 32 en la página 316](#).

2. Compruebe si la red es lenta.

Si los mensajes no se mueven lo bastante rápido a través de un canal, puede ser debido a que la red sea lenta. Para identificar si éste es el caso, realice las verificaciones siguientes:

a) En la salida del mandato `DIS CHSTATUS(QM1.TO.QM2) ALL`, compruebe el campo NETTIME.

Estos indicadores se miden cuando el canal emisor solicita a su socio una respuesta. Esto sucede al final de cada proceso por lotes y, cuando un canal está desocupado durante las pulsaciones.

b) Si este indicador muestra que los viajes de ida y vuelta tardan más de lo esperado, utilice otras herramientas de supervisión de red para investigar el rendimiento de la red.

3. Compruebe si el canal utiliza la compresión.

Si el canal utiliza la compresión, esto se suma al tiempo empleado para procesar los mensajes. Si el canal utiliza sólo un algoritmo de compresión, efectúe las comprobaciones siguientes:

a) En la salida del mandato `DIS CHSTATUS(QM1.TO.QM2) ALL`, compruebe el campo COMPTIME.

Estos indicadores muestran el tiempo empleado durante la compresión o la descompresión.

- b) Si la compresión elegida no reduce la cantidad de datos que deben enviarse en la cantidad esperada, cambie el algoritmo de compresión.
4. Si el canal utiliza varios algoritmos de compresión, efectúe las comprobaciones siguientes:
- a) En la salida del mandato DIS CHSTATUS(QM1 . TO . QM2) ALL, compruebe los campos COMPTIME, COMPHDR y COMPMSG.
 - b) Cambie los algoritmos de compresión especificados en la definición de canal, o considere la posibilidad de escribir una salida de mensajes para alterar temporalmente la elección de algoritmo de compresión del canal para mensajes concretos, si el índice de compresión o la elección del algoritmo, no proporciona la compresión o el rendimiento necesarios.

Resolver problemas con canales de clúster

Si ha apreciado un aumento en el número de mensajes de la cola SYSTEM.CLUSTER.TRANSMIT.QUEUE, el primer paso que debe efectuar para diagnosticar el problema es descubrir qué canal, o canales, tienen problemas con la entrega de mensajes.

Acerca de esta tarea

Descubrir qué canal, o canales, que utilicen la cola SYSTEM.CLUSTER.TRANSMIT.QUEUE tienen un problema con la entrega de mensajes. Realice las comprobaciones siguientes:

Procedimiento

1. Emita el mandato siguiente:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Nota: Si tiene un clúster ocupado en el que haya muchos mensajes en movimiento, considere la posibilidad de emitir este mandato, con un número superior, para poder eliminar los canales que tengan sólo unos pocos mensajes disponibles que se deban entregar.

2. Consulte detenidamente la salida del canal, o canales, que tengan valores grandes en el campo XQMSGSA. Determine por qué el canal no mueve los mensajes, o no los mueve con una rapidez suficiente. Utilice las tareas descritas en [“Supervisión de canales”](#) en la [página 312](#) para diagnosticar los problemas surgidos con los canales que provoquen el aumento de mensajes.

Supervisión de clústeres

Dentro de un clúster puede supervisar mensajes de aplicaciones, mensajes de control y registros. Existen consideraciones de supervisión especiales cuando la carga del clúster se equilibra entre dos o más instancias de una cola.

Supervisión de mensajes de aplicación en el clúster

Normalmente, todos los mensajes de clúster que abandonan el gestor de colas pasan a través del SYSTEM.CLUSTER.TRANSMIT.QUEUE, independientemente del canal emisor del clúster que se esté utilizando para transmitir el mensaje. Cada canal drena los mensajes destinados a dicho canal en paralelo con los demás canales de emisor de clúster. Una acumulación de mensajes cada vez mayor en esta cola puede indicar un problema con uno o varios canales, y debe investigarse:

- La profundidad de la cola debe ser supervisada adecuadamente para el diseño del clúster.
- El siguiente mandato devuelve todos los canales que tengan más de un mensaje en espera en la cola de transmisión:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Con todos los mensajes de clúster en una misma cola, no siempre resulta fácil ver qué canal tiene problemas cuando empieza a llenarse. El uso de este mandato es un método fácil para ver qué canal es el responsable.

Puede configurar un gestor de colas para que tenga varios gestores de transmisión. Si cambia el atributo de gestor de colas DEFCLXQ por CHANNEL, cada canal de clúster emisor está asociado a una cola de transmisión de clúster diferente. O bien, puede configurar manualmente colas de transmisión por separado. Para visualizar todas las colas de transmisión del clúster que están asociadas a canales de clúster emisor, ejecute el mandato:

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

Defina colas de transmisión de clúster de modo que sigan el patrón de tener la raíz fija del nombre de cola a la izquierda. A continuación, puede consultar la profundidad de todas las colas de transmisión de clúster devueltas por el mandato **DISPLAY CLUSMGR** , utilizando un nombre de cola genérico:

```
DISPLAY QUEUE (qname *) CURDEPTH
```

Supervisión de mensajes de control en el clúster

La cola de SYSTEM . CLUSTER . COMMAND . QUEUE se utiliza para procesar todos los mensajes de control de clúster para un gestor de colas, generados por el gestor de colas local o enviados a este gestor de colas desde otros gestores de colas del clúster. Cuando un gestor de colas mantiene correctamente su estado de clúster, esta cola tiende a cero. Hay casos en los que la profundidad de los mensajes en esta cola puede aumentar temporalmente, aunque:

- Tener muchos mensajes en la cola indica pérdidas en el estado del clúster.
- Cuando se realizan cambios significativos, deje que la cola se adapte entre ellos. Por ejemplo, al trasladar repositorios, deje que la cola llegue a cero antes de trasladar el segundo repositorio.

Mientras exista un retraso de mensajes en esta cola, las actualizaciones del estado del clúster o los mandatos relacionados con el clúster no se procesan. Si los mensajes no se eliminan de esta cola durante mucho tiempo, es necesaria una investigación adicional, inicialmente a través de la inspección de los registros de errores del gestor de colas (o registros de CHINIT en z/OS), lo que podría explicar el proceso que está causando esta situación.

SYSTEM . CLUSTER . REPOSITORY . QUEUE contiene la información de la memoria caché del repositorio de clúster como un número de mensajes. Es normal que haya siempre mensajes en esta cola, y aún más si el clúster es grande. Por lo tanto, la profundidad de los mensajes en esta cola no es un tema de preocupación.

Supervisión de registros

Es posible que los problemas que se producen en el clúster no muestren síntomas externos en las aplicaciones durante varios días (e incluso meses) después del inicio del problema, debido al almacenamiento en la memoria caché de la información y a la naturaleza distribuida de los clústeres. No obstante, el problema original se suele notificar en los registros de errores de IBM MQ (y registros de CHINIT en z/OS). Por este motivo, es fundamental supervisar activamente estos registros para ver los mensajes relacionados con los clústeres. Estos mensajes deben leerse atentamente y, cuando sea necesario, se deben tomar medidas correctivas.

Por ejemplo: una interrupción de las comunicaciones con un gestor de colas en un clúster puede revelar que determinados recursos del clúster se están suprimiendo debido a la forma en que los clústeres revalidan periódicamente los recursos de clúster publicando de nuevo la información. Un aviso de que se está produciendo potencialmente un suceso de este tipo se indica mediante el mensaje [AMQ9465](#) o [CSQX465I](#) en sistemas z/OS. Este mensaje indica que el problema debe investigarse.

Consideraciones especiales relativas al equilibrio de carga

Cuando la carga de clúster se equilibra entre dos o más instancias de una cola, las aplicaciones consumidoras deben procesar mensajes en cada una de las instancias. Si una o varias de esas aplicaciones consumidoras termina o deja de procesar mensajes, es posible que la agrupación en clúster

continúe enviando mensajes a las instancias de la cola. En este caso, los mensajes no se procesan hasta que las aplicaciones vuelvan a funcionar correctamente. Por este motivo, la supervisión de las aplicaciones es una parte importante de la solución y se deben tomar medidas para redireccionar los mensajes en ese caso. Puede encontrar un ejemplo de un mecanismo para automatizar esta supervisión en: [El programa de ejemplo de supervisión de colas de clúster \(AMQSCLM\)](#).

Conceptos relacionados

[“Ajuste de redes de publicación/suscripción distribuidas” en la página 382](#)

Utilice los consejos de ajuste de esta sección para ayudar a mejorar el rendimiento de sus clústeres y jerarquías de publicación/suscripción distribuida de IBM MQ.

[“Equilibrio entre productos y consumidores en las redes de publicación/suscripción” en la página 388](#)

Un concepto importante en el rendimiento de la mensajería asíncrona es el *equilibrio*. A menos que los consumidores de mensajes estén equilibrados con los productores de mensajes, existe el riesgo de que se acumule el retraso de los mensajes no consumidos y afecte gravemente al rendimiento de varias aplicaciones.

Supervisión de la conmutación de colas de transmisión

Es importante que supervise el proceso de conmutación de colas de transmisión de canales de clúster emisor para minimizar el impacto en la empresa. Por ejemplo, no debe intentar este proceso cuando la carga de trabajo es alta o conmutando muchos canales simultáneamente.

El proceso de conmutación de canales

El proceso utilizado para conmutar canales es:

1. El canal abre la nueva cola de transmisión para entrada y empieza a obtener mensajes de ella (utilizando obtener por ID de correlación)
2. El gestor de colas inicia un proceso en segundo plano para mover los mensajes en cola para el canal desde su cola de transmisión antigua a su nueva cola de transmisión. Mientras los mensajes se están moviendo, los mensajes nuevos para el canal se ponen en cola en la cola de transmisión antigua para conservar la secuenciación. Este proceso puede tardar un tiempo en completarse si hay un gran número de mensajes para el canal en su cola de transmisión antigua, o si llegan mensajes nuevos rápidamente.
3. Cuando ningún mensaje confirmado o no confirmado permanece en cola para el canal en su cola de transmisión antigua, el conmutador se completa. Los nuevos mensajes se colocan ahora directamente en la nueva cola de transmisión.

Para evitar la eventualidad de que varios canales conmuten simultáneamente, IBM MQ proporciona la posibilidad de conmutar la cola de transmisión de uno o varios canales que no se están ejecutando. En:

- IBM MQ for Multiplatforms el mandato se denomina **runswchl**
- IBM MQ for z/OS se puede utilizar el programa de utilidad CSQUTIL para procesar un mandato SWITCH CHANNEL en su lugar

Supervisión del estado de las operaciones de conmutador

Para comprender el estado de las operaciones de conmutación, los administradores pueden realizar las acciones siguientes:

- Supervise el registro de errores del gestor de colas (AMQERR01.LOG) donde se generan los mensajes para indicar las etapas siguientes durante la operación:
 - La operación de conmutación se ha iniciado
 - Se ha iniciado el movimiento de mensajes
 - Actualizaciones periódicas sobre cuántos mensajes quedan por mover (si la operación de conmutación no se completa rápidamente)
 - El movimiento de mensajes se ha completado

- La operación de conmutación se ha completado

En z/OS, estos mensajes se envían a las anotaciones de trabajo del gestor de colas, no a las anotaciones de trabajo del iniciador de canal, aunque un canal envía un único mensaje a las anotaciones de trabajo del iniciador de canal si inicia una conmutación al iniciar.

- Utilice el mandato `DISPLAY CLUSQMGR` para consultar la cola de transmisión que está utilizando actualmente cada canal de clúster emisor.
- Ejecute el mandato `runswch1` (o `CSQUTIL` en z/OS) en modalidad de consulta para determinar el estado de conmutación de uno o varios canales. La salida de este mandato identifica lo siguiente para cada canal:
 - Si el canal tiene una operación de conmutación pendiente
 - Qué cola de transmisión está conmutando el canal de y a
 - ¿Cuántos mensajes permanecen en la cola de transmisión antigua?

Cada mandato es realmente útil, porque en una invocación puede determinar el estado de cada canal, el impacto que ha tenido un cambio de configuración y si se han completado todas las operaciones de conmutación.

Problemas potenciales que pueden producirse

Consulte [Problemas potenciales al conmutar colas de transmisión](#) para obtener una lista de algunos problemas que se pueden encontrar al conmutar la cola de transmisión, sus causas y las soluciones más probables.

Conceptos relacionados

“[Ajuste de redes de publicación/suscripción distribuidas](#)” en la página 382

Utilice los consejos de ajuste de esta sección para ayudar a mejorar el rendimiento de sus clústeres y jerarquías de publicación/suscripción distribuida de IBM MQ.

“[Equilibrio entre productos y consumidores en las redes de publicación/suscripción](#)” en la página 388

Un concepto importante en el rendimiento de la mensajería asíncrona es el *equilibrio*. A menos que los consumidores de mensajes estén equilibrados con los productores de mensajes, existe el riesgo de que se acumule el retraso de los mensajes no consumidos y afecte gravemente al rendimiento de varias aplicaciones.

Multi

Supervisión del equilibrio de aplicaciones

Puede utilizar el mandato `DISPLAY APSTATUS` para supervisar el estado del equilibrio de aplicaciones a través de un clúster uniforme, e investigar por qué la aplicación no está equilibrada si esto no es lo esperado.

Supervisión del estado actual de las aplicaciones entre gestores de colas de un clúster

Desde cualquier gestor de colas de un clúster uniforme puede obtener un estado actual de las aplicaciones en todos los gestores de colas de un clúster ejecutando el mandato `DIS APSTATUS`.

A partir de IBM MQ 9.2.0, el campo **TYPE** también se visualiza en la salida.

Por ejemplo, después de que el gestor de colas se haya iniciado, puede ver una salida como la siguiente:

```

1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)

```

Esto muestra que hay una aplicación llamada MYAPP en el clúster uniforme, y actualmente hay ocho instancias, las cuales se consideran movibles alrededor del clúster uniforme. El valor equilibrado de UNKNOWN es un valor temporal, que indica que el gestor de colas aún no ha sido objeto de un intento de reequilibrio de la aplicación si es necesario.

Después de un breve período de tiempo, es más probable que vea la salida siguiente:

```

1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)

```

Esta salida muestra que la aplicación tiene ocho instancias, pero que no están equilibradas a través del clúster uniforme. En este punto, vale la pena examinar la distribución de las aplicaciones en el clúster.

Para ello, vuelva a ejecutar el mandato **DIS APSTATUS**. Tenga en cuenta que puede ejecutar este mandato en cualquier gestor de colas en el clúster uniforme:

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                      MOVCCOUNT(6)
BALSTATE(HIGH)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)            QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                      MOVCCOUNT(1)
BALSTATE(LOW)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)            QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                      MOVCCOUNT(1)
BALSTATE(LOW)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)            QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

A partir de este punto, puede ver en este momento que el gestor de colas UNID001 tiene seis instancias, pero los gestores de colas UNID0002 y UNID0003 sólo tienen una cada uno. La salida de BALSTATE proporciona una indicación del estado del equilibrio la última vez que el gestor de colas lo notificó. Sin embargo, tenga en cuenta que el recuento de instancias puede ser más reciente que el campo BALSTATE.

Esta salida también proporciona una buena indicación de que el clúster uniforme está distribuyendo información sobre esta aplicación. La aplicación es conocida por todas las instancias del gestor de colas en el clúster uniforme, y la fecha y hora del último mensaje es muy reciente.

Además, el campo ACTIVE indica que se considera que todos los gestores de colas en el clúster se están comunicando entre sí; si ACTIVE se establece en NO en cualquier gestor de colas, indica que la comunicación regular con él se han interrumpido.

Si se deja que se reequilibre por sí mismo, el resultado acabaría siendo algo así como:

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                      MOVCCOUNT(3)
BALSTATE(OK)                   LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)            QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                      MOVCCOUNT(3)
BALSTATE(OK)                   LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)            QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)

```

```

AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(2)                      MOVCOUNT(2)
BALSTATE(OK)                   LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)            QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

El BALSTATE de los gestores de colas es ahora Correcto lo que indica que se ha alcanzado un estado estable.

Supervisión de instancias de aplicación individuales

Puede ver las instancias de aplicación individuales, pero esto se hace para cada gestor de colas. Examinando la salida en UNID001:

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                      IMMREASN(NONE)
IMMCOUNT(0)                   IMMDATE( )
IMMTIME( )                    MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                      IMMREASN(NONE)
IMMCOUNT(0)                   IMMDATE( )
IMMTIME( )                    MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                      IMMREASN(NONE)
IMMCOUNT(0)                   IMMDATE( )
IMMTIME( )                    MOVABLE(YES)
TYPE (LOCAL)
One MQSC command read.

```

La presencia de MOVABLE(YES) indica que esta instancia se puede mover si es necesario a otro gestor de colas en el clúster. En el siguiente ejemplo, la instancia de aplicación no se puede mover porque no está conectada como un cliente:

```

3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1)                      IMMREASN(NOTCLIENT)
IMMCOUNT(0)                   IMMDATE( )
IMMTIME( )                    MOVABLE(NO)
TYPE (LOCAL)

```

Puede utilizar el campo **CONNTAG** para ver las conexiones del gestor de colas individuales de dicha instancia, lo que puede ser útil si la instancia de la aplicación es IMMREASN(NOTRECONN). En el ejemplo siguiente, la aplicación cliente no se puede mover y la investigación de las opciones de conexión muestra que no tiene MQCNO_RECONNECT en el campo **CONNOPTS**.

```

1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1)                      IMMREASN(NOTRECONN)
IMMCOUNT(0)                   IMMDATE( )
IMMTIME( )                    MOVABLE(NO)
TYPE (LOCAL)

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)

```

```

TYPE(CONN)
PID(14656)
APPLDESC(IBM MQ Channel)
APPLTYPE(USER)
CHANNEL(SYSTEM.DEF.SVRCONN)
CONNNAME(127.0.0.1)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserId)
UOWSTDA( )
UOWLOGDA( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
TYPE(CONN)
TID(20)
APPLTAG(ClientApp)
ASTATE(NONE)
CLIENTID( )
UOWLOG( )
UOWSTTI( )
UOWLOGTI( )
UOWSTATE(NONE)

```

Conceptos relacionados

[Equilibrio de aplicaciones automático](#)

Monitoring performance and resource usage on z/OS

Use this topic to understand the facilities available to monitor the performance, and resource usage of your IBM MQ for z/OS subsystems.

Related tasks

[Configuring queue managers on z/OS](#)

[Administering IBM MQ for z/OS](#)

Introduction to monitoring IBM MQ for z/OS

Use this topic as an overview of the monitoring facilities available for IBM MQ for z/OS. For example, obtaining snapshots, using IBM MQ trace, online monitoring, and events.

This topic describes how to monitor the performance and resource usage of IBM MQ.

- It outlines some of the information that you can retrieve and briefly describes a general approach to investigating performance problems. See [“Investigating performance problems” on page 330](#) for more information.
- It describes how you can collect statistics about the performance of IBM MQ by using SMF records.
- It describes how to gather accounting data to enable you to charge your customers for their use of your IBM MQ systems.
- It describes how to use IBM MQ events (alerts) to monitor your systems.

Here are some of the tools you might use to monitor IBM MQ; they are described in the sections that follow:

- Tools provided by IBM MQ:
 - [Using DISPLAY commands](#)
 - [“Using CICS adapter statistics” on page 326](#)
 - [“Using IBM MQ events” on page 327](#)
- z/OS service aids:
 - [“Using System Management Facility” on page 328](#)
- Other IBM licensed programs:
 - [Using the Resource Measurement Facility](#)
 - [Using Tivoli Decision Support for z/OS](#)
 - [Using the CICS monitoring facility](#)

Information about interpreting the data gathered by the performance statistics trace is given in [“Interpreting IBM MQ for z/OS performance statistics” on page 331](#).

Information about interpreting the data gathered by the accounting trace is given in “[Interpreting IBM MQ for z/OS accounting data](#)” on page 368.

Getting snapshots of IBM MQ using the DISPLAY commands

IBM MQ provides the MQSC facility which can give a snapshot of the performance, and resource usage using the DISPLAY commands.

You can get an idea of the current state of IBM MQ by using the DISPLAY commands and, for the CICS adapter, the CICS adapter panels.

Using DISPLAY commands

You can use the IBM MQ MQSC DISPLAY or PCF Inquire commands to obtain information about the current state of IBM MQ. They provide information about the status of the command server, process definitions, queues, the queue manager, and its associated components. These commands are:

MQSC command	PCF command
DISPLAY ARCHIVE	Inquire Archive
DISPLAY AUTHINFO	Inquire Authentication Information Object
DISPLAY CFSTATUS	Inquire CF Structure Status
DISPLAY CFSTRUCT	Inquire CF Structure
DISPLAY CHANNEL	Inquire Channel
DISPLAY CHINIT	Inquire Channel Initiator
DISPLAY CHSTATUS	Inquire Channel Status
DISPLAY CMDSERV	
DISPLAY CLUSQMGR	Inquire Cluster Queue Manager
DISPLAY CONN	Inquire Connection
DISPLAY GROUP	Inquire Group
DISPLAY LOG	Inquire Log
DISPLAY PROCESS	Inquire Process
DISPLAY QMGR	Inquire Queue Manager
DISPLAY QSTATUS	Inquire Queue Status
DISPLAY QUEUE	Inquire Queue
DISPLAY SECURITY	Inquire Security
DISPLAY STGCLASS	Inquire Storage Class
DISPLAY SYSTEM	Inquire System
DISPLAY TRACE	
DISPLAY USAGE	Inquire Usage

For the detailed syntax of each command, see [MQSC commands](#) or [PCF commands](#). All of the functions of these commands (except DISPLAY CMDSERV and DISPLAY TRACE) are also available through the operations and control panels.

These commands provide a snapshot of the system only at the moment the command was processed. If you want to examine trends in the system, you must start an IBM MQ trace and analyze the results over a period of time.

Using CICS adapter statistics

If you are an authorized CICS user, you can use the CICS adapter control panels to display CICS adapter statistics dynamically.

These statistics provide a snapshot of information related to CICS thread usage and situations when all threads are busy. The display connection panel can be refreshed by pressing the Enter key.

For more information about configuring the IBM MQ CICS adapter, see the [Configuring connections to MQ](#) section of the CICS documentation.

Using IBM MQ trace

You can record performance statistics and accounting data for IBM MQ by using the IBM MQ trace facility. Use this topic to understand how to control IBM MQ trace.

The data generated by IBM MQ is sent to:

- The System Management Facility (SMF), specifically as SMF record type 115, subtypes 1 and 2 for the performance statistics trace
- The SMF, specifically as SMF record type 116, subtypes zero, 1, and 2 for the accounting trace.


If you prefer, the data generated by the IBM MQ accounting trace can also be sent to the generalized trace facility (GTF).

Starting IBM MQ trace

You can start the IBM MQ trace facility at any time by issuing the IBM MQ [START TRACE](#) command.


Accounting data can be lost if the accounting trace is started or stopped while applications are running. To collect accounting data successfully, the following conditions must apply:

- The accounting trace must be active when an application starts, and it must still be active when the application finishes.
- If the accounting trace is stopped, any accounting data collection that was active stops.

You can also start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro.  These parameters are described in [Using CSQ6SYSP](#).

Before starting an IBM MQ trace, read [“Using System Management Facility” on page 328](#).

Controlling IBM MQ trace

To control the IBM MQ trace data collection at start-up, specify values for the parameters in the CSQ6SYSP macro when you customize IBM MQ.  See [Using CSQ6SYSP](#) for details.

You can control IBM MQ tracing when the queue manager is running with these commands:

- START TRACE
- ALTER TRACE
- STOP TRACE

You can choose the destination to which trace data is sent. Possible destinations are:

SMF

System Management Facility

GTF

Generalized Trace Facility (accounting trace only)

SRV

Serviceability routine for diagnostic use by IBM service personnel

For daily monitoring, information is sent to SMF (the default destination). SMF data sets typically contain information from other systems; this information is not available for reporting until the SMF data set is dumped.

You can also send accounting trace information to the GTF. This information has an event identifier of 5EE.

z/OS The `MQI` call and user parameter, and `z/OS` generalized trace facility (GTF) describes how to deal with IBM MQ trace information sent to the GTF.

For information about IBM MQ commands, see [MQSC commands](#).

Effect of trace on IBM MQ performance

Using the IBM MQ trace facility can have a significant effect on IBM MQ and transaction performance. For example, if you start a global trace for class 1 or for all classes, it is likely to increase processor usage and transaction response times by approximately 50%. However, if you start a global trace for classes 2 - 4 alone, the increase in processor usage and transaction response times is likely to be less than 1% additional processor cost to the cost of IBM MQ calls. The same applies for a statistics or accounting trace.

z/OS Using IBM MQ online monitoring

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the `MONQ`, `MONCHL`, and `MONACLS` attributes.

Table 33 on page 327 summarizes the commands to set these attributes at different levels and to display the monitoring information.

Attribute	Applicable at this level	Set using command	Display monitoring information using command
MONQ	Queue	DEFINE QLOCAL DEFINE QMODEL ALTER QLOCAL ALTER QMODEL	DISPLAY QSTATUS
	Queue manager	ALTER QMGR	
MONCHL	Channel	DEFINE CHANNEL ALTER CHANNEL	DISPLAY CHSTATUS
	Queue manager	ALTER QMGR	
MONACLS	Queue manager	ALTER QMGR	

For full details of these commands, see [MQSC commands](#). For more information about online monitoring, see [“Supervisión de la red de IBM MQ” on page 5](#).

z/OS Using IBM MQ events

IBM MQ instrumentation events provide information about errors, warnings, and other significant occurrences in a queue manager. You can monitor the operation of all your queue managers by incorporating these events into your own system management application.

IBM MQ instrumentation events fall into the following categories:

Queue manager events

These events are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist.

Performance events

These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached, or the queue was not serviced within a predefined time limit.

Channel events

These events are reported by channels as a result of conditions detected during their operation. For example, a channel instance is stopped.

Configuration events

These events are notifications that an object has been created, changed, or deleted.

When an event occurs, the queue manager puts an *event message* on the appropriate *event queue*, if defined. The event message contains information about the event that can be retrieved by a suitable IBM MQ application.

IBM MQ events can be enabled using the IBM MQ commands or the operations and control panels.

See “Tipos de suceso” on page 12 for information about the IBM MQ events that generate messages, and for information about the format of these messages. See [Event message reference](#) for information about enabling the events.

Using System Management Facility

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

System management facility (SMF) is a z/OS service aid used to collect information from various z/OS subsystems. This information is dumped and reported periodically, for example, hourly. You can use SMF with the IBM MQ trace facility to collect data from IBM MQ. In this way you can monitor *trends*, for example, in system utilization and performance, and collect accounting information about each user ID using IBM MQ.

To record performance statistics (record type 115) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(115))
```

To record accounting information (record type 116) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(116))
```

To use the z/OS command SETSMF, either PROMPT(ALL) or PROMPT(LIST) must be specified in the SMFPRM xx member. See [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) for more information.

You can start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro; this is described in [Using CSQ6SYSP](#).

Specifying YES on the SMFSTAT and SMFACCT parameters enables you to collect trace information as a queue manager starts.

You can also start collection of the data when the queue manager is running with the **START TRACE** command, specifying START TRACE(A) or START TRACE(S).

You can turn on or off the recording of accounting information at the queue or queue manager level using the ACCTQ parameter of the **DEFINE QLOCAL**, **DEFINE QMODEL**, **ALTER QLOCAL**, **ALTER QMODEL**, or **ALTER QMGR** commands. See [MQSC commands](#) for details of these commands.

You can control the collection of channel accounting data at the channel or queue manager level using the **STATCHL** parameter of the **DEFINE CHANNEL**, **ALTER CHANNEL** or **ALTER QMGR** commands.

You can specify the interval at which IBM MQ collects statistics and accounting data in one of these ways:

- You can collect statistics data and accounting data at different intervals, using STATIME (statistics data) and ACCTIME (accounting data) in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data at the same interval by specifying a value for STATIME in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data by specifying zero for STATIME.

SMF must be running before you can send data to it. For more information about SMF, see the [z/OS MVS System Management Facilities \(SMF\)](#) manual.

For the statistics and accounting data to be reset, at least one MQI call must be issued during the accounting interval.

Allocating additional SMF buffers

When you start a trace, you must ensure that you allocate adequate SMF buffers. Specify SMF buffering on the VSAM BUFSP parameter of the access method services DEFINE CLUSTER statement. Specify CISZ(4096) and BUFSP(81920) on the **DEFINE CLUSTER** statement for each SMF VSAM data set.

If an SMF buffer shortage occurs, SMF rejects any trace records sent to it. IBM MQ sends a CSQW133I message to the z/OS console when this occurs. IBM MQ treats the error as temporary and remains active even though SMF data can be lost. When the shortage has been alleviated and trace recording has resumed, IBM MQ sends a CSQW123I message to the z/OS console.

Reporting data in SMF

You can use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed.

There are several ways to report on this data, for example:

- Write an application program to read and report information from the SMF data set. You can then tailor the report to fit your exact needs.
- Use Performance Reporter to process the records. For more information, see [“Using other products with IBM MQ”](#) on page 330.

No interval CLASS(03) SMF accounting records produced during long running processes

You are collecting CLASS(3) SMF116 accounting records for IBM MQ, but are getting no records produced while a long running process runs.

The CLASS(3) SMF116 accounting records normally are produced only when a process ends. For long running processes, for example CICS, this might not produce a sufficient number of records, as the process can run for a month or longer. However, you might want to gather SMF116 records at set time intervals while a process is running.

To gather CLASS(3) SMF116 accounting records you must set the following:

SMFACCT

=YES

SMFSTAT

=YES or NO, where

YES

Causes records to be produced if a collection broadcast is received.

No

Causes you to get a CLASS(3) SMF116 record produced only when a process ends

and issue the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(03)
```

If you have set SMFSTAT=YES and a collection broadcast occurs, an interval CLASS(3) SMF116 accounting record is produced for any process currently running that was also running at the time of the previous collection broadcast.

You can set the collection broadcast to occur on a regular time interval by setting STATIME in [CSQ6SYSP](#) as follows:

- If your STATIME has been set to a value greater than 0, that is your broadcast interval in minutes.
- If your STATIME = 0 the SMF broadcast of your system is used (SMF INTVAL)
- If your STATIME = 0 and your SMF INTVAL is not set, no broadcast occurs and no interval records are produced

Using other products with IBM MQ

You can use other products to help you to improve the presentation of, or to augment statistics related to, performance and accounting. For example, Resource Measurement Facility, Tivoli Decision Support, and CICS monitoring.

Using the Resource Measurement Facility

Resource Measurement Facility (RMF) is an IBM licensed program (program number 5685-029) that provides system-wide information about processor utilization, I/O activity, storage, and paging. You can use RMF to monitor the utilization of physical resources across the whole system dynamically. For more information, see the [z/OS Resource Measurement Facility User's Guide](#).

Using Tivoli Decision Support for z/OS

You can use Tivoli Decision Support for z/OS to interpret RMF and SMF records.

Tivoli Decision Support for z/OS is an IBM licensed program (program number 5698-B06) that enables you to manage the performance of your system by collecting performance data in a Db2® database and presenting the data in various formats for use in systems management. Tivoli Decision Support for can generate graphic and tabular reports using systems management data it stores in its Db2 database. It includes an administration dialog, a reporting dialog, and a log collector, all of which interact with a standard Db2 database.

This is described in the [IBM Tivoli Decision Support for z/OS: Administration Guide and Reference](#).

Using the CICS monitoring facility

The CICS monitoring facility provides performance information about each CICS transaction running. It can be used to investigate the resources used and the time spent processing transactions. For background information, see the [CICS Performance Guide](#) and [Developing CICS System Programs](#), together with the two companion reference manuals, formerly called the [CICS Customization Guide](#).

Investigating performance problems

Performance problems can arise from various factors. For example, incorrect resource allocation, poor application design, and I/O restraints. Use this topic to investigate some of the possible causes of performance problems.

Performance can be adversely affected by:

- Buffer pools that are an incorrect size
- Lack of real storage
- I/O contention for page sets or logs

- Log buffer thresholds that are set incorrectly
- Incorrect setting of the number of log buffers
- Large messages
- Units of recovery that last a long time, incorporating many messages for each sync point
- Messages that remain on a queue for a long time
- RACF® auditing
- Unnecessary security checks
- Inefficient program design

When you analyze performance data, always start by looking at the overall system before you decide that you have a specific IBM MQ problem. Remember that almost all symptoms of reduced performance are magnified when there is contention. For example, if there is contention for DASD, transaction response times can increase. Also, the more transactions there are in the system, the greater the processor usage and greater the demand for both virtual and real storage.

In such situations, the system shows heavy use of *all* its resources. However, the system is actually experiencing normal system stress, and this stress might be hiding the cause of a performance reduction. To find the cause of such a loss of performance, you must consider all items that might be affecting your active tasks.

Investigating the overall system

Within IBM MQ, the performance problem is either increased response time or an unexpected and unexplained heavy use of resources. First check factors such as total processor usage, DASD activity, and paging. An IBM tool for checking total processor usage is resource management facility (RMF). In general, you must look at the system in some detail to see why tasks are progressing slowly, or why a specific resource is being heavily used.


Start by looking at general task activity, then focus on particular activities, such as specific tasks or a specific time interval.

Another possibility is that the system has limited real storage; therefore, because of paging interrupts, the tasks progress more slowly than expected.

Investigating individual tasks

You can use the accounting trace to gather information about IBM MQ tasks. These trace records tell you a great deal about the activity that the task has performed, and about how much time the task spent suspended, waiting for latches. The trace record also includes information about how much Db2 and coupling facility activity were performed by the task.

Interpreting IBM MQ accounting data is described in [“Interpreting IBM MQ for z/OS accounting data” on page 368](#).

Long running units of work can be identified by the presence of message CSQR026I in the job log. This message indicates that a task has existed for more than three queue manager checkpoints and its log records have been shunted.  For a description of log record shunting, see [The log files](#).

Interpreting IBM MQ for z/OS performance statistics

Use this topic as an index to the different SMF records created by IBM MQ for z/OS.

IBM MQ for z/OS performance statistics are written as SMF type 115 records. Statistics records are produced periodically at a time interval specified by the **STATIME** parameter of the CSQ6SYSP system parameter module, or at the SMF global recording interval if you specify zero for **STATIME**. The information provided in the SMF records comes from the following components of IBM MQ:

Buffer manager	Manages the buffer pools in virtual storage and the writing of pages to page sets as the buffer pools become full. Also manages the reading of pages from page sets.
Coupling facility manager	Manages the interface with the coupling facility.
Data manager	Manages the links between messages and queues. It calls the buffer manager to process the pages with messages on them.
Db2 manager	Manages the interface with the Db2 database that is used as the shared repository.
Lock manager	Manages locks
Log manager	Manages the writing of log records, which are essential for maintaining the integrity of the system if there is a back out request, or for recovery, if there is a system or media failure.
Message manager	Processes all IBM MQ API requests.
Storage manager	Manages storage, for example, storage pool allocation, expansion, and deallocation.
Topic manager	Manages the topic and subscription information
Coupling facility SMDS manager	Manages the shared message data sets (SMDS) for large messages stored in the coupling facility.

IBM MQ statistics are written to SMF as SMF type 115 records. The following subtypes can be present:

1

System information, for example, related to the logs and storage.

2

Information about number of messages and paging information. Queue sharing group information related to the coupling facility and Db2.

5 and 6

Detailed information about internal storage usage in the queue manager address space. While you can view this information, some of it is intended only for IBM use.

7

Storage manager summary information. While you can view this information, some of it is intended only for IBM use.

201

Page set input/output information

215

Buffer pool information

216

Queue information

231

System information for the channel initiator address space.

Note that:

- Subtype 1, 2, 201, and 215 records are created with statistics trace class 1.
- Subtype 5, 6, and 7 records are created with statistics trace class 3.
- Subtype 231 records are created with statistics trace class 4.
- Subtype 216 records are created with statistics trace class 5.

The subtype is specified in the SM115STF field (shown in [Table 34 on page 336](#)).

Utilice este tema como referencia para los comandos necesarios para recopilar registros SMF de tipo 115 y 116.

Usando el comando INICIAR SEGUIMIENTO

Utilice la versión dinámica del INICIAR SEGUIMIENTO comando con las siguientes opciones para recopilar los registros:

- `START TRACE(STAT) DEST SMF CLASS(*)` y `START TRACE(ACCTG) DEST SMF CLASS(*)` inicia rastreos para las clases 1 a 3
- `START TRACE(STAT) DEST SMF CLASS(4)` y `START TRACE(ACCTG) DEST SMF CLASS(4)` inicia las estadísticas del iniciador del canal y los datos de contabilidad del canal respectivamente.

Ver Planificación de datos SMF del iniciador de canal para obtener más detalles sobre la información del iniciador del canal de clase 4.

- `START TRACE(STAT) DEST SMF CLASS(5)` inicia las estadísticas de la cola.



Atención: Si está utilizando la versión de consola del comando agregue `cpf` al inicio del comando que emita.

Para las estadísticas del canal `CLASS(4)`, debe configurar el atributo `STATCHL` en la definición `CHANNEL`. Para obtener más información, consulte “Control de la recopilación de estadísticas de canal” en la página 147.

Para las estadísticas de cola `CLASS(5)`, debe establecer el atributo `STATQ` en la definición `QUEUE` y/o `QMGR`. Para obtener más información, consulte “Control de la recopilación de estadísticas de cola” en la página 146.

Registros SMF tipo 115 -IBM MQ Estadísticas

Para verificar la recopilación de estadísticas actuales, emita el MOSTRAR TRAZA dominio `DISPLAY TRACE(STAT)` . Deberá ver lo siguiente:

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
02 STAT 01,02,03,04 SMF * *
END OF TRACE REPORT
```

Nota: Establecer el atributo `ZPARM SMFSTAT` o `ZPARM SMFACCT` en un asterisco ahora solo controla las clases 1, 2 y 3. Debe activar las clases 4 y 5 usando el comando `START TRACE`.

Datos SMF tipo 116 Clase 3 y 4 –IBM MQ registros contables de tareas y canales

Para verificar la recopilación de estadísticas actuales, emita el MOSTRAR TRAZA dominio `DISPLAY TRACE(ACCT)` . Deberá ver lo siguiente:

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
03 ACCTG * SMF * *
END OF TRACE REPORT
CSQ9022I QML1 CSQWVCM1 'DISPLAY TRACE' NORMAL COMPLETION
```

Usted puede dar vuelta `SMF116` La recopilación de datos se activa y desactiva dinámicamente, ya que no es necesario ningún ciclo del administrador de colas, utilizando los siguientes comandos:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
START TRACE(ACCTG) DEST(SMF) CLASS(4)
```

```
STOP TRACE(ACCTG) DEST(SMF) CLASS(3)
STOP TRACE(ACCTG) DEST(SMF) CLASS(4)
```

Los datos de contabilidad del canal requieren una configuración adicional para producir el SMF. Los pasos son los siguientes:

- Emita el comando para activar la recopilación de información para todas las conexiones del remitente, receptor y cliente:

```
ALTER QMGR STATCHL(HIGH)
```

- Emita el comando para activar la recopilación de información para canales de clúster definidos automáticamente si la agrupación está en uso:

```
ALTER QMGR STATCLS(HIGH)
```

Nota: En las declaraciones anteriores, el valor ALTO, MEDIO o BAJO tiene el mismo efecto.

- Verifique que todos los canales tengan STATCHL configurado en QMGR:

```
DISPLAY CHANNEL(*) STATCHL
```

Verificando el STATIME



Atención: Debe establecer el intervalo en no más de cinco minutos para cumplir con las últimas herramientas, utilizando el [CONFIGURAR EL SISTEMA](#) comando, aunque el valor predeterminado es 30 minutos.

Si está evaluando un grupo de uso compartido de colas (QSG), asegúrese de que STATIME sea coherente en todo el QSG.

- Muestre la configuración del sistema usando lo siguiente [SISTEMA DE VISUALIZACIÓN](#) dominio:

```
DISPLAY SYSTEM
```

- Si el valor de STATIME es cero, lo que significa que el intervalo de estadísticas está establecido en el valor LPAR predeterminado, normalmente esto está bien.

Ha habido casos de LPAR de muy alto volumen que crearon duraciones variables de la producción de datos SMF. Si en su evaluación de los datos ha habido duraciones muy variables, establezca STATIME en un valor distinto de cero para todos los gestores de colas.

- Los siguientes ejemplos establecen el intervalo STATIME en cinco minutos:

```
SET SYSTEM STATIME(05)
```

o si en IBM MQ 9.2.4 o superior:

```
SET SYSTEM STATIME(05.00)
```

Nota: El nuevo intervalo no entra en vigor hasta que haya transcurrido el intervalo actual, por lo que debe realizar el cambio antes de comenzar la recopilación de datos.

Verificación del valor ACCTIME para gestores de colas que están en el nivel de versión IBM MQ 9.2.4 o mas alto:

- Si ACCTIME se establece en -1, es lo mismo que el atributo STATIME.
- Si esta evaluación es para un QSG, asegúrese de que ACCTIME sea coherente en todo el QSG. Si no es así, establezca los valores para que sean el mismo intervalo antes de comenzar a recopilar los datos.

- Si ACCTTIME está configurado en 30 o más, modifíquelo para que no sea mayor que 15 usando el [CONFIGURAR EL SISTEMA](#) dominio. Los siguientes ejemplos establecen el intervalo ACCTIME en cinco minutos:

```
SET SYSTEM ACCTIME(05.00)
```

Nota: El nuevo intervalo no entra en vigor hasta que haya transcurrido el intervalo actual, por lo que debe realizar el cambio antes de comenzar la recopilación de datos.

Verificación de la configuración ACCTQ en los administradores de colas

- Muestre la configuración ACCTQ para los gestores de colas que está investigando utilizando el siguiente comando:

```
DISPLAY QMGR ACCTQ
```

- Si el valor es ACCTQ(ON), no es necesaria ninguna otra acción. De lo contrario, emita el siguiente comando:

```
ALTER QMGR ACCTQ(ON)
```

- Si se utiliza la agrupación en clústeres, asegúrese de que la contabilidad esté activada para elSYSTEM.CLUSTER.TRANSMIT.QUEUE y cualquier otro clúster con nombre que transmita colas alojadas por el gestor de colas. Para determinar si la contabilidad está habilitada para las colas de transmisión del clúster:

- Muestre la configuración ACCTQ para la cola o colas, usando el siguiente comando:

```
DISPLAY QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ
```

- Si el valor es ACCTQ(ON), no es necesaria ninguna otra acción. De lo contrario, emita el siguiente comando:

```
ALTER QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ(ON)
```

- Para todas las demás colas de gran volumen, verifique que el valor de ACCTQ esté activado o establezca ACCTG en QMGR.

Layout of an SMF type 115 record

You can use this section as a reference for the format of an SMF type 115 record.

The standard layout for SMF records involves three parts:

SMF header

Provides format, identification, and time and date information about the record itself.

Self-defining section

Defines the location and size of the individual data records within the SMF record.

Data records

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

Related reference

[“The SMF header” on page 336](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 336](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 337](#)

Use this topic to understand some example SMF records.

z/OS The SMF header

Use this topic as a reference for the format of the SMF header.

Table 34 on page 336 shows the format of SMF record header (SM115).

Table 34. SMF record 115 header description

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM115	SMF record header.	
0	0	Integer	2	SM115LEN	SMF record length.	14A0
2	2		2		Reserved.	
4	4	Integer	1	SM115FLG	System indicator.	5E
5	5	Integer	1	SM115RTY	Record type. The SMF record type, for IBM MQ statistics records this is always 115 (X'73').	73
6	6	Integer	4	SM115TME	Time when SMF moved record.	00355575
10	A	Integer	4	SM115DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM115SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM115SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM115STF	Record subtype.	0002
24	18	Character	3	SM115REL	IBM MQ version.	F6F0F0 (600)
27	1B		1		Reserved	
28	1C	Character	0	SM115END	End of SMF header and start of self-defining section.	

z/OS Self-defining sections

Use this topic as a reference for format of the self-defining sections of the SMF record.

A self-defining section of a type 115 SMF record tells you where to find a statistics record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at fixed offsets from the start of the SMF record. Each statistics record can be identified by an eye-catcher string.

The following types of self-defining section are available to users for type 115 records. Each self-defining section points to statistics data related to one of the IBM MQ components. Table 35 on page 336 summarizes the sources of the statistics, the eye-catcher strings, and the offsets of the self-defining sections from the start of the SMF record header.

Table 35. Offsets to self-defining sections

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Eye-catcher of data
		Dec	Hex	
Storage manager	1	100	X'64'	QSST

Table 35. Offsets to self-defining sections (continued)

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Eye-catcher of data
		Dec	Hex	
Log manager	1	116	X'74'	QJST
Message manager	2	36	X'24'	QMST
Data manager	2	44	X'2C'	QIST
No longer used. The self-defining section will be binary zeros.	2	52	X'34'	
Lock manager	2	60	X'3C'	QLST
Db2 manager	2	68	X'44'	Q5ST
Coupling Facility manager	2	76	X'4C'	QEST
Topic manager	2	84	X'54'	QTST
SMDS usage	2	92	X'5C'	QESD
Buffer manager - one for each buffer pool	215	36	X'24'	QPST
Channel initiator	231			QWSX
Data manager page set - one for each page set	201	36	X'24'	QIS1
Storage manager	5	36	X'24'	QSPH
Storage manager	6	36	X'24'	QSGM
Storage manager	7	36	X'24'	QSRS
Queues	216	36	X'24'	QQST

Note: Some of the storage manager information in subtype 5, 6 and 7 records is intended only for IBM use. Other self-defining sections that are not listed contain data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssllllnnnn
```

where:

- ssssssss is a fullword containing the offset from the start of the SMF record.
- llll is a halfword giving the length of this data record.
- nnnn is a halfword giving the number of data records in this SMF record.

For more information see, “Examples of SMF statistics records” on page 337.

Note: Always use offsets in the self-defining sections to locate the statistics records.

Examples of SMF statistics records

Use this topic to understand some example SMF records.

Figure 20 on page 338 shows an example of part of the SMF record for subtype 1. Subtype 1 includes the storage manager and log manager statistics records. The SMF record header is shown underlined.

The self-defining section at offset X'64' refers to storage manager statistics and the self-defining section at offset X'74' refers to log manager statistics, both shown in **bold**.

The storage manager statistics record is located at offset X'0000011C' from the start of the header and is X'48' bytes long. There is one set of storage manager statistics, identified by the eye-catcher string QSST. The start of this statistics record is also shown in the example.

The log manager statistics record is located at offset X'00000164' from the start of the header and is X'78' bytes long. There is one set of log manager statistics, identified by the eye-catcher string QJST.

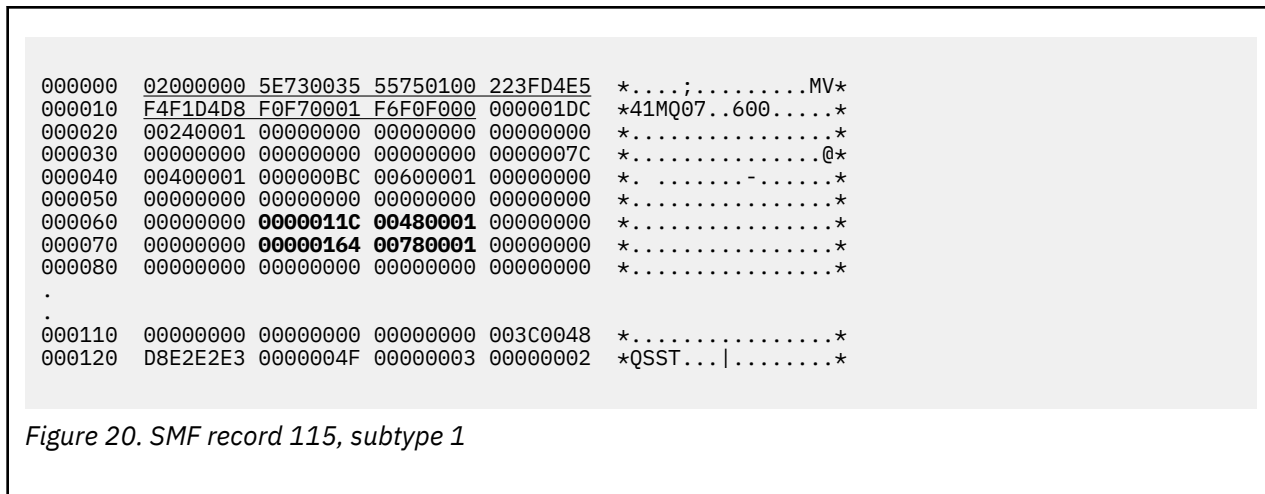


Figure 21 on page 339 shows an example of part of the SMF record for subtype 2. Subtype 2 includes the statistics records for the message, data, lock, coupling facility, topic, and Db2 managers. The SMF record header is shown underlined; the self-defining sections are shown alternately **bold** and *italic*.

- The self-defining section at offset X'24' refers to message manager statistics. The message manager statistics record is located at offset X'00000064' from the start of the header and is X'48' bytes long. There is one set of these statistics, identified by the eye-catcher string QMST.
- The self-defining section at offset X'2C' refers to data manager statistics. The data manager statistics record is located at offset X'000000AC' from the start of the header and is X'50' bytes long. There is one set of these statistics, identified by the eye-catcher string QIST.
- In releases prior to IBM MQ 9.1.0, the self-defining section at offset X'34' referred to buffer manager statistics. As this SMF record was taken from an IBM MQ 9.1.0 queue manager, the buffer manager self-defining section is set to zeros to indicate that there are no buffer manager statistics. Instead, these statistics are in SMF 115 subtype 215 records.
- The self-defining section at offset X'3C' refers to lock manager statistics. The lock manager statistics record is located at offset X'000000FC' from the start of the header and is X'20' bytes long. There is one set of these statistics, identified by the eye-catcher string QLST.
- The self-defining section at offset X'44' refers to Db2 manager statistics. The Db2 manager statistics record is located at offset X'0000011C' from the start of the header and is X'2A0' bytes long. There is one set of these statistics, identified by the eye-catcher string Q5ST.
- The self-defining section at offset X'4C' refers to coupling facility manager statistics. The coupling facility manager statistics record is located at offset X'000003BC' from the start of the header and is X'1008' bytes long. There is one set of these statistics, identified by the eye-catcher string QEST.
- The self-defining section at offset X'54' refers to topic manager statistics. The topic manager statistics record is located at offset X'000013C4' from the start of the header and is X'64' bytes long. There is one set of these statistics, identified by the eye-catcher string QTST.
- The self-defining section at offset X'5C' is for SMDS statistics. This self-defining section is set to zeros indicating that SMDS is not being used.

```

000000 09F40000 5E730033 4DBE0113 142FD4E5 *.4..;...(. ....MV*
000010 F4F1D4D8 F2F10002 F9F1F000 00001428 *41MQ21..910.....*
000020 00240001 00000064 00480001 000000AC *.....*
000030 00500001 00000000 00000000 000000FC *.....*
000040 00200001 0000011C 02A00001 000003BC *.....*
000050 10080001 000013C4 00640001 00000000 *.....D.....*
000060 00000000 D40F0048 D8D4E2E3 00000000 *...M...QMST...*
000080 00000000 00000000 00000000 00000000 *.....*
000090 00000000 00000000 00000000 00000000 *.....*
0000A0 00000000 00000000 00000000 C90F0050 *.....I..&*
0000B0 D8C9E2E3 00000000 00000000 00000000 *QIST.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 D30F0020 *.....L...*
000100 D8D3E2E3 00000000 00000000 00000000 *QLST.....*
000110 00000000 00000000 00000000 F50F02A0 *.....5...*
000120 D8F5E2E3 00000008 00000000 00000000 *Q5ST.....*
.

```

Figure 21. SMF record 115, subtype 2

Processing type 115 SMF records

Use this topic as a reference for processing type 115 SMF records.

You must process any data you collect from SMF to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the values of the following fields:

- SM115RTY, the SMF record number, must be X'73' (115)
- SM115STF, the record subtype, must be 0001, 0002, 0005, 0006, 0007, 0201, 0215, or 0231

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility”](#) on page 328.

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

Storage manager data records

Use this topic as a reference for storage manager data records.

The format of the storage manager statistics record is described in assembler macro thlqual.SCSQMACS(CSQDQSST).

The data contains information about the number of fixed and variable storage pools that the queue manager has allocated, expanded, contracted, and deleted during the statistics interval, plus the number of GETMAIN, FREEMAIN, and STORAGE requests to z/OS, including a count of those requests that were unsuccessful. Additional information includes a count of the number of times the short-on-storage condition was detected and a count of the number of abends that occurred as a result of that condition.

Additional data about storage usage in the queue manager is produced by class 2 and class 3 statistics trace. While you can view this information, some of it is intended only for IBM use.

- The format of the storage manager pool header statistics record, which is present in subtype 5 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSPH).
- The format of the storage manager getmain statistics record, which is present in subtype 6 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSGM).

- The format of the storage manager region summary record, which is present in subtype 7 records, is described in assembler macro `thlqual.SCSQMACS(CSQDQSR)`.

Registros de datos del gestor de registros

Utilice este tema como referencia para registros de datos del gestor de registros.

El formato del registro de estadísticas del gestor de anotaciones cronológicas se describe en la macro de ensamblador `thlqual.SCSQMACS(CSQDQJST)`.

En las estadísticas, estos recuentos son importantes:

1. El número total de solicitudes de grabación de registro:

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. El número total de solicitudes de lectura de registro:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

Los síntomas del problema que se pueden examinar con las estadísticas del gestor de registro que se describen en la siguiente tabla.

<p>Síntoma 1 QJSTWTB no es cero.</p> <p>Razón Las tareas se suspenden mientras que el almacenamiento intermedio en almacenamiento se graba en el registro activo. Puede surgir problemas al grabar en el registro activo. El parámetro OUTBUFF dentro de CSQ6LOGP es demasiado pequeño.</p> <p>Acción Investigue los problemas que hay al grabar en el registro activo. Aumente el valor del parámetro OUTBUFF dentro de CSQ6LOGP.</p>
<p>Síntoma 2 La proporción: $\text{QJSTWTL} / N_{\text{logread}}$ es mayor que 1%.</p> <p>Razón Se iniciaron lecturas del registro que tenían que leer de un registro de archivado, pero IBM MQ no ha podido asignar un conjunto de datos porque los conjuntos de datos MAXRTU ya estaban asignados.</p> <p>Acción Aumente MAXRTU.</p>

Síntoma 3

La proporción: $QJSTRARH/N_{logread}$ es mayor de lo normal.

Razón

La mayoría de las solicitudes de lectura de registro deben proceder del almacenamiento intermedio de salida o del registro activo. Para satisfacer las solicitudes de restitución, los registros de la unidad de recuperación se leen del almacenamiento intermedio, el registro activo y los registros archivados.

Una unidad de recuperación de larga ejecución, que se extiende a lo largo de un período de muchos minutos, puede tener registros de anotaciones repartidos entre muchos registros distintos. Esto degrada el rendimiento porque se tiene que llevar a cabo trabajo adicional para recuperar los registros de anotaciones.

Acción

Cambie la aplicación para reducir la longitud de una unidad de recuperación. Además, piense en aumentar el tamaño del registro activo para reducir la posibilidad de que una única unidad de recuperación se extienda a más de un registro.

Otros punteros

La proporción $N_{logread} / N_{logwrite}$ proporciona una indicación de la cantidad de trabajo que se debe restituir.

Síntoma 4

$QJSTLLCP$ es más de 10 por hora.

Razón

En un sistema con mucha actividad, normalmente cabría esperar ver 10 puntos de comprobación por hora. Si el valor $QJSTLLCP$ es mayor que este, indica un problema en la configuración del gestor de colas.

La razón más probable para ello es que el parámetro $LOGLOAD$ en $CSQ6SYSP$ es demasiado pequeño. El otro suceso que provoca un punto de comprobación es cuando un registro activo se llena y pasa al siguiente conjunto de datos de registro activo. Si los registros son demasiado pequeños, esto puede causar puntos de comprobación frecuentes.

$QJSTLLCP$ es un recuento del número total de puntos de comprobación.

Acción

Aumente el parámetro de $LOGLOAD$, o aumente el tamaño de los conjuntos de datos de registro, según sea necesario.

Síntoma 5

$QJSTCmpFail > 0$ o $QJSTCmpComp$ no mucho menor que $QJSTCmpUncmp$

Razón

El gestor de colas no puede comprimir significativamente los registros de anotaciones.

$QJSTCmpFail$ es el número de veces que el gestor de colas no ha podido conseguir alguna reducción de la longitud de registro. Debe comparar el número con $QJSTCmpReq$ (número de solicitudes de compresión) para ver si el número de anomalías es significativa.

$QJSTCmpComp$ es el total de bytes comprimidos grabados en el registro y $QJSTCmpUncmp$ es el total de bytes antes de la compresión. Ningún total contiene los bytes grabados para los registros de anotaciones que no fueron elegibles para su compresión. Si los números son similares, la compresión ha logrado pocos beneficios.

Acción

Desactive la compresión del registro. Emita el mandato `SET LOG COMPLOG(NONE)`. Consulte el mandato [SET LOG](#) para obtener más información.

Nota: En el primer conjunto de estadísticas producidas después de inicio del sistema, puede haber una actividad de registro significativa debido a la resolución de unidades de recuperación en curso.

Cambios en zHyperWrite

V 9.4.0

A partir de IBM MQ 9.3.5, el proceso de grabación de zHyper altera el comportamiento de las estadísticas QJSTHWC y QJSTHWE.

Antes de IBM MQ 9.3.5:

- QJSTHWC es el número de conjuntos de datos de registro utilizados en el intervalo de SMF que están en volúmenes con capacidad de grabación de zHyper. La capacidad de grabación de zHyper de un volumen se obtiene durante el inicio del gestor de colas y puede cambiar con el tiempo, por lo que esta información puede estar obsoleta.
- QJSTHWE es el número de conjuntos de datos de registro utilizados en el intervalo de SMF en los que se graban con zHyperWrite habilitado. Esto sucede si el conjunto de datos de registro está en un volumen con capacidad de grabación de zHyper y si el gestor de colas se ha habilitado para la grabación de zHyper estableciendo ZHYWRITE (YES).

Desde IBM MQ 9.3.5:

- QJSTHWC es el número de conjuntos de datos de registro utilizados en el intervalo de SMF que están en volúmenes con capacidad de grabación de zHyper. La capacidad de grabación de zHyper de un volumen se obtiene durante el inicio del gestor de colas y puede cambiar con el tiempo, por lo que esta información puede estar obsoleta.
- QJSTHWE es el número de conjuntos de datos de registro utilizados en el intervalo de SMF en los que se graban con zHyperWrite habilitado. Esto sucede si el gestor de colas se ha habilitado para zHyperWrite estableciendo ZHYWRITE (YES).



Atención: La capacidad de grabación de zHyper de un volumen puede cambiar con el tiempo. A partir de IBM MQ 9.3.5, esto puede dar lugar a escenarios en los que QJSTHWE es mayor que QJSTHWC.

Estadísticas de enlace de zHyper

V 9.4.0

A partir de IBM MQ 9.4.0, se han añadido estadísticas del gestor de registros para realizar un seguimiento del rendimiento de zHyperLink.

Se han añadido las estadísticas siguientes a QJST:

Nombre	Tipo	Bytes de longitud	Descripción
QJSTHLSCIW	Entero sin signo	4	Número de grabaciones de página única que han solicitado el enlace zHyper.
QJSTHLMCIW	Entero sin signo	4	Número de grabaciones de varias páginas que han solicitado el enlace zHyper.
QJSTHLICNTW	Entero sin signo	4	Número de páginas grabadas donde la grabación ha solicitado el enlace zHyper.
QJSTHLSCIS	Entero sin signo	4	Número de grabaciones de página que han utilizado correctamente el enlace zHyper. Es decir, la escritura se hizo de forma síncrona.
QJSTHLMCIS	Entero sin signo	4	Número de grabaciones de varias páginas que han utilizado correctamente el enlace zHyper. Es decir, la escritura se hizo de forma síncrona.
QJSTHLICNTS	Entero sin signo	4	Número de páginas grabadas correctamente utilizando el enlace zHyper. Es decir, la escritura se hizo de forma síncrona.

Nombre	Tipo	Bytes de longitud	Descripción
QJSTHLSCIF	Entero sin signo	4	Número de grabaciones de página única que han intentado utilizar el enlace zHyper, pero la grabación no se ha podido realizar de forma síncrona. En su lugar, la grabación se ha realizado correctamente de forma asíncrona.
QJSTHLMCIF	Entero sin signo	4	Número de grabaciones de varias páginas que han intentado utilizar el enlace zHyper, pero la grabación no se ha podido realizar de forma síncrona. En su lugar, la grabación se ha realizado correctamente de forma asíncrona.
QJSTHLICNTF	Entero sin signo	4	Número de páginas grabadas donde la grabación ha intentado utilizar el enlace zHyper, pero la grabación no se ha podido realizar de forma síncrona. En su lugar, la grabación se ha realizado correctamente de forma asíncrona.
QJSTHLSCICONT	Entero sin signo	4	Número de grabaciones de página única antes de que se estableciera una sesión de grabación de enlace zHyper. Consulte la nota “1” en la página 344 .
QJSTHLMCICONT	Entero sin signo	4	Número de grabaciones de varias páginas antes de que se estableciera una sesión de grabación de enlace zHyper. Consulte la nota “1” en la página 344 .
QJSTHLCONCICNT	Entero sin signo	4	Número de páginas grabadas antes de que se estableciera una conexión de enlace zHyper. Consulte la nota “1” en la página 344 .
QJSTHLSCIWTMAX	Entero sin signo	8	Tiempo de grabación de enlace zHyper más largo en intervalo, página única.
QJSTHLSCIWTMIN	Entero sin signo	8	Tiempo de grabación de enlace zHyper más corto en intervalo, página única.
QJSTHLSCIWTTOT	Entero sin signo	8	Tiempo total de grabación de enlace zHyper en intervalo, página única.
QJSTHLMCIWTMAX	Entero sin signo	8	Tiempo de grabación de enlace zHyper más largo en intervalo, varias páginas.
QJSTHLMCIWTMIN	Entero sin signo	8	Tiempo de grabación de enlace zHyper más corto en intervalo, varias páginas.
QJSTHLMCIWTTOT	Entero sin signo	8	Tiempo total de grabación de enlace zHyper en intervalo, varias páginas.
QJSTHLIOSQU	Entero sin signo	16	Suma de cuadrados de tiempos de grabación de enlace de zHyper, para escrituras de página única.
QJSTHLC	Entero sin signo	4	Número de registros nuevos, utilizados en este intervalo de SMF, que tienen capacidad de enlace zHyper.
QJSTHLE	Entero sin signo	4	Número de registros nuevos, utilizados en este intervalo de SMF, que están habilitados para el enlace zHyper.

Nota:

1. Al cambiar a una nueva copia de registro activo, puede que haya un periodo de tiempo antes de que se establezca una sesión de grabación de enlace zHypercon el DASD.

z/OS Message manager data records

Use this topic as a reference for message manager data records.

The format of the message manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQMST)`.

The data gives you counts of different IBM MQ API requests.

z/OS Data manager data records

Use this topic as a reference for the format of the Data Manager data records.

The format of the data manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIST)`.

The data gives you counts of different object requests.

z/OS Data manager page set data records

Use this section as a reference for the format of the data manager page set data records

The format of the data manager page set statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIS1)`.

The page set usage information helps to facilitate better management of local queues within a queue manager by recording information such as page set input-output rates and highest usage.

The data provides the same basic page set information as output by the MQSC **DISPLAY USAGE TYPE (PAGESET)** command, or the PCF Inquire Usage (**MQCMD_INQUIRE_USAGE**) command.

For example:

- The total pages
- The current used pages
- Unused persistent and nonpersistent pages
- Expansion method
- Number of extends
- Number of stripes

The data also provides some performance indicators, together with other performance information. For example:

- How many times deferred write, immediate write, and read page I/O requests happened during the SMF interval,
- The number of pages moved, elapsed time and number of read and write operations.
- How many pages written in checkpoints.
- Has expansion occurred during the SMF interval?
- How many times the page set became full.
- An indication of where new space is being allocated within the page set.

From the information displayed, you should be able to understand the general status of each page set, and consider whether you need to retune the system.

Related reference

[“The SMF header” on page 336](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 336](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 337](#)

Use this topic to understand some example SMF records.

Buffer manager data records

Use this topic as a reference for the format of buffer manager data records.

The format of the buffer manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQPST)`.

Note: Buffer manager statistics records will only be created for buffer pools that are defined. If a buffer pool is defined but not used then no values will be set and its buffer manager statistics record will not contain any data.

For information about efficiently managing your buffer pools, see [“Managing your buffer pools” on page 346](#).

When interpreting the statistics, you are recommended to consider the following factors because the values of these fields can be used to improve the performance of your system:

1. If QPSTSOS, QPSTDMC, or QPSTIMW is greater than zero, you should either increase the size of the buffer pool or reallocate the page sets to different buffer pools.
 - QPSTSOS is the number of times that there were no buffers available for page get requests. If QPSTSOS ever becomes nonzero, it shows that IBM MQ is under severe stress. The buffer pool size should be increased. If increasing the buffer pool size does not make the value of QPSTSOS zero, there might be I/O contention on the DASD page sets.
 - QPSTDMC is the number of updates that were performed synchronously because there was either more than 95% of the pages in the buffer pool waiting for write I/O, or there was less than 5% of the buffer pool available for read requests. If this number is not zero, the buffer pool might be too small and should be enlarged. If increasing the buffer pool size does not reduce QPSTDMC to zero, there might be I/O contention on the DASD page sets.
 - QPSTIMW is a count of the number of times pages were written out synchronously. If QPSTDMC is zero, QPSTIMW is the number of times pages were found on the queue waiting for write I/O that had been there for at least two checkpoints.
2. For buffer pool zero and buffer pools that contain short-lived messages:
 - QPSTDWT should be zero, and the percentage QPSTCBSL/QPSTNBUF should be greater than 15%.

QPSTDWT is the number of times the asynchronous write processor was started because there was either more than 85% of the pages in the buffer pool waiting for write I/O, or there was less than 15% of the buffer pool available for read requests. Increasing the buffer pool size should reduce this value. If it does not, the pattern of access is one of long delays between puts and gets.
 - QPSTTPW might be greater than zero due to checkpointing activity.
 - QPSTRIO should be zero unless messages are being read from a page set after the queue manager is restarted.

The ratio of QPSTRIO to QPSTGETP shows the efficiency of page retrieval within the buffer pool. Increasing the buffer pool size should decrease this ratio and, therefore, increase the page retrieval efficiency. If this does not happen, it indicates that pages are not being frequently reaccessed. This implies a transaction pattern where there is a long delay between messages being put and then later retrieved.

The ratio of QPSTGETN to QPSTGETP indicates the number of times an empty page, as opposed to a non-empty page, has been requested. This ratio is more an indication of transaction pattern, than a value that can be used to tune the system.

- If QPSTSTL has a value greater than zero, this indicates that pages that have not been used before are now being used. This might be caused by an increased message rate, messages not being processed as fast as they were previously (leading to a buildup of messages), or larger messages being used.

QPSTSTL is a count of the number of times a page access request did not find the page already in the buffer pool. Again, the lower the ratio of QPSTSTL to (QPSTGETP + QPSTGETN) is, the higher the page retrieval efficiency. Increasing the buffer pool size should decrease this ratio but, if it does not, it is an indication that there are long delays between puts and gets.

- You are recommended to have sufficient buffers to handle your peak message rate.
3. For buffer pools with long-lived messages, where there are more messages than can fit into the buffer pool:
- (QPSTRIO+QPSTWIO)/Statistics interval is the I/O rate to page sets. If this value is high, you should consider using multiple page sets on different volumes to allow I/O to be carried out in parallel.
 - Over the period of time that the messages are processed (for example, if messages are written to a queue during the day and processed overnight) the number of read I/Os (QPSTRIO) should be approximately the total number of pages written (QPSTTPW). This shows that one page is read for every page written.

If QPSTRIO is much larger than QPSTTPW, this shows that pages are being read in multiple times. This might be a result of the application using MQGET by *MsgId* or *CorrelId* when the queue is not indexed, or browsing messages on the queue using get next.

The following actions might relieve this problem:

- a. Increase the size of the buffer pool so that there are enough pages to hold the queue, in addition to any changed pages.
- b. Use the INDXTYPE queue attribute, which allows a queue to be indexed by *MsgId* or *CorrelId* and eliminates the need for a sequential scan of the queue.
- c. Change the design of the application to eliminate the use of MQGET with *MsgId* or *CorrelId*, or the get next with browse option.

Note: Applications using long-lived messages typically process the first available message and do not use MQGET with *MsgId* or *CorrelId*, and they might browse only the first available message.

- d. Move page sets to a different buffer pool to reduce contention between messages from different applications.

Managing your buffer pools

To manage your buffer pools efficiently, you must consider the factors that affect the buffer pool I/O operations and also the statistics associated with the buffer pools.

The following factors affect buffer pool I/O operations.

- If a page containing the required data is not found in the buffer pool, it is read in synchronously to an available buffer from its DASD page set.
- Whenever a page is updated, it is put on an internal queue of pages to be (potentially) written out to DASD. This means that the buffer used by that page is unavailable for use by any other page until the buffer has been written to DASD.
- If the number of pages queued to be written to DASD exceeds 85% of the total number of buffers in the pool, an asynchronous write processor is started to put the buffers to DASD.

Similarly, if the number of buffers available for page get requests become less than 15% of the total number of buffers in the pool, the asynchronous write processor is started to perform the write I/O operations.

The write processor stops when the number of pages queued to be written to DASD has fallen to 75% of the total number of buffers in the pool.

- If the number of pages queued for writing to DASD exceeds 95% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

Similarly, if the number of buffers available for page get requests becomes less than 5% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

- If the number of buffers available for page get requests ever reaches zero, a transaction that encounters this condition is suspended until the asynchronous write processor has finished.
- If a page is frequently updated, the page spends most of its time on the queue of pages waiting to be written to DASD. Because this queue is in least recently used order, it is possible that a frequently updated page placed on this least recently used queue is never written out to DASD. For this reason, at the time of update, if the page is found to have been waiting on the write operation to DASD queue for at least two checkpoints, it is synchronously written to DASD. Updating occurs at checkpoint time and is suspended until the asynchronous write processor has finished.

The aim of this algorithm is to maximize the time pages spend in buffer pool memory while allowing the system to function if the system load puts the buffer pool usage under stress.

z/OS Lock manager data records

Use this topic as a reference to the format of the lock manager data records.

The format of the lock manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQLST)`.

The records contain data about the following information:

- The number of lock get requests and lock release requests.
- The number of times a lock get request determined that the requested lock was already held.

z/OS Db2 manager data records

Use this topic as a reference to the format of the Db2 manager data records.

The format of the Db2 manager statistics record is described in the following table and in assembler macro `thlqual.SCSQMACS(CSQDQ5ST)` and C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `q5st`, `q5stid`.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	668	Q5ST	Db2 manager statistics
0	0	Bitstring	2	Q5STID	Control block identifier
2	2	Integer	2	Q5STLL	Control block length
4	4	Character	4	Q5STEYEC	Control block eye catcher
8	8	Character	660	Q5STZERO	QMST part cleared on occasion
8	8	Integer	4	NUMTASK	Number of server tasks
12	C	Integer	4	ACTTASK	Number of active server tasks
16	10	Integer	4	CONNCNT	Number of connect requests
20	14	Integer	4	DISCCNT	Number of disconnect requests
24	18	Integer	4	DHIGMAX	Max. request queue depth
28	1C	Integer	4	ABNDCNT	Number of Db2SRV task abends
32	20	Integer	4	REQUCNT	Number of requests requeued
36	24	Integer	4	DEADCNT	Number of deadlock timeouts
40	28	Integer	4	DELECNT	Number of delete requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
44	2C	Integer	4	LISTCNT	Number of list requests
48	30	Integer	4	READCNT	Number of read requests
52	34	Integer	4	UPDCNT	Number of update requests
56	38	Integer	4	WRITCNT	Number of write requests
60	3C	Integer	4	SCSSEL	SCST (shared-channel-status) selects
64	40	Integer	4	SCSINS	SCST inserts
68	44	Integer	4	SCSUPD	SCST updates
72	48	Integer	4	SCSDEL	SCST deletes
76	4C	Integer	4	SSKSEL	SSKT (shared-sync-key) selects
80	50	Integer	4	SSKINS	SSKT inserts
84	54	Integer	4	SSKDEL	SSKT deletes
88	58	Integer	4	SCSBFTS	SCST number of times buffer too small
92	5C	Integer	4	SCSMAXR	SCST maximum rows on query
96	60	Integer	4	* (2)	Reserved
104	68	Character	8	DELETCUW	Cumulative STCK difference - Thread delete
112	70	Character	8	DELETMXW	Maximum STCK difference - Thread delete
120	78	Character	8	DELESCUW	Cumulative STCK difference - SQL delete
128	80	Character	8	DELESMXW	Maximum STCK difference - SQL delete
136	88	Character	8	LISTTCUW	Cumulative STCK difference - Thread list
144	90	Character	8	LISTTMXW	Maximum STCK difference - Thread list
152	98	Character	8	LISTSCUW	Cumulative STCK difference - SQL list
160	A0	Character	8	LISTSMXW	Maximum STCK difference - SQL list
168	A8	Character	8	READTCUW	Cumulative STCK difference - Thread read
176	B0	Character	8	READTMXW	Maximum STCK difference - Thread read
184	B8	Character	8	READSCUW	Cumulative STCK difference - SQL read
192	C0	Character	8	READSMXW	Maximum STCK difference - SQL read
200	C8	Character	8	UPDTTCUW	Cumulative STCK difference - Thread update
208	D0	Character	8	UPDTTMXW	Maximum STCK difference - Thread update
216	D8	Character	8	UPDTSCUW	Cumulative STCK difference - SQL update
224	E0	Character	8	UPDTSMXW	Maximum STCK difference - SQL update
232	E8	Character	8	WRITTCUW	Cumulative STCK difference - Thread write
240	F0	Character	8	WRITTMXW	Maximum STCK difference - Thread write
248	F8	Character	8	WRITSCUW	Cumulative STCK difference - SQL write
256	100	Character	8	WRITSMXW	Maximum STCK difference - SQL write

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
264	108	Character	8	SCSSTCUW	Cumulative STCK difference - Thread select
272	110	Character	8	SCSSTMXW	Maximum STCK difference - Thread select
280	118	Character	8	SCSSSCUW	Cumulative STCK difference - SQL select
288	120	Character	8	SCSSSMXW	Maximum STCK difference - SQL select
296	128	Character	8	SCSITCUW	Cumulative STCK difference - Thread insert
304	130	Character	8	SCSITMXW	Maximum STCK difference - Thread insert
312	138	Character	8	SCSISCUW	Cumulative STCK difference - SQL insert
320	140	Character	8	SCSISMXW	Maximum STCK difference - SQL insert
328	148	Character	8	SCSUTCUW	Cumulative STCK difference - Thread update
336	150	Character	8	SCSUTMXW	Maximum STCK difference - Thread update
344	158	Character	8	SCSUSCUW	Cumulative STCK difference - SQL update
352	160	Character	8	SCSUSMXW	Maximum STCK difference - SQL update
360	168	Character	8	SCSDTCUW	Cumulative STCK difference - Thread delete
368	170	Character	8	SCSDTMXW	Maximum STCK difference - Thread delete
376	178	Character	8	SCSDSCUW	Cumulative STCK difference - SQL delete
384	180	Character	8	SCSDSMXW	Maximum STCK difference - SQL delete
392	188	Character	8	SSKSTCUW	Cumulative STCK difference - Thread select
400	190	Character	8	SSKSTMXW	Maximum STCK difference - Thread select
408	198	Character	8	SSKSSCUW	Cumulative STCK difference - SQL select
416	1A0	Character	8	SSKSSMXW	Maximum STCK difference - SQL select
424	1A8	Character	8	SSKITCUW	Cumulative STCK difference - Thread insert
432	1B0	Character	8	SSKITMXW	Maximum STCK difference - Thread insert
440	1B8	Character	8	SSKISCUW	Cumulative STCK difference - SQL insert
448	1C0	Character	8	SSKISMXW	Maximum STCK difference - SQL insert
456	1C8	Character	8	SSKDTCUW	Cumulative STCK difference - Thread delete
464	1D0	Character	8	SSKDTMXW	Maximum STCK difference - Thread delete
472	1D8	Character	8	SSKDSCUW	Cumulative STCK difference - SQL delete
480	1E0	Character	8	SSKDSMXW	Maximum STCK difference - SQL delete
488	1E8	Integer	4	LMSSEL	Number of Db2 BLOB read requests
492	1EC	Integer	4	LMSINS	Number of Db2 BLOB insert requests
496	1F0	Integer	4	LMSUPD	Number of Db2 BLOB update requests
500	1F4	Integer	4	LMSDEL	Number of Db2 BLOB delete requests
504	1F8	Integer	4	LMSLIS	Number of Db2 BLOB list requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
508	IFC	64 bit integer	8	LMSSTCUW	Total elapsed time for all thread read BLOB requests
516	204	64 bit integer	8	LMSSTMXW	Maximum elapsed time for a thread read BLOB request
524	20C	64 bit integer	8	LMSSSCUW	Total elapsed time for all SQL read BLOB requests
532	214	64 bit integer	8	LMSSSMXW	Maximum elapsed time for an SQL read BLOB request
540	21C	64 bit integer	8	LMSITCUW	Total elapsed time for all thread insert BLOB requests
548	224	64 bit integer	8	LMSITMXW	Maximum elapsed time for a thread insert BLOB request
556	22C	64 bit integer	8	LMSISCUW	Total elapsed time for all SQL insert BLOB requests
564	234	64 bit integer	8	LMSISMXW	Maximum elapsed time for an SQL insert BLOB request
572	23C	64 bit integer	8	LMSUTCW	Total elapsed time for all thread update BLOB requests
580	244	64 bit integer	8	LMSUTMXW	Maximum elapsed time for a thread update BLOB request
588	24C	64 bit integer	8	LMSUSCUW	Total elapsed time for all SQL update BLOB requests
596	254	64 bit integer	8	LMSUSMXW	Maximum elapsed time for an SQL update BLOB request
604	25C	64 bit integer	8	LMSDTCW	Total elapsed time for all thread delete BLOB requests
612	264	64 bit integer	8	LMSDTMXW	Maximum elapsed time for a thread delete BLOB request
620	26C	64 bit integer	8	LMSDSCW	Total elapsed time for all SQL delete BLOB requests
628	274	64 bit integer	8	LMSDSMXW	Maximum elapsed time for an SQL delete BLOB request
636	27C	64 bit integer	8	LMSLTCW	Total elapsed time for all thread list BLOB requests
644	284	64 bit integer	8	LMSLTMXW	Maximum elapsed time for a thread list BLOB request
652	28C	64 bit integer	8	LMSLSCW	Total elapsed time for all SQL list BLOB requests
660	294	64 bit integer	8	LMSLSMXW	Maximum elapsed time for an SQL list BLOB request

The data contains counts for each request type that the Db2 resource manager supports. For these request types, maximum and cumulative elapse times are kept for the following:

- The time spent in the Db2 resource manager as a whole (called the thread time).
- The time that was spent performing the RRSAF and SQL parts of the request (a subset of the thread time called the SQL time).

Information is also provided for:

- The number of server tasks attached.
- The maximum overall request depth against any of the server tasks.
- The number of times any of the server task requests terminated abnormally.

If the abnormal termination count is not zero, a requeue count is provided indicating the number of queued requests that were requeued to other server tasks as a result of the abnormal termination.

If the average thread time is significantly greater than the average SQL time, this might indicate that thread requests are spending an excessive amount of time waiting for a server task to process the SQL part of the request. If this is the case, examine the DHIGMAX field and, if the value is greater than one, consider increasing the number of Db2 server tasks specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro.

Coupling facility manager data records

Use this topic as a reference to the format of the coupling facility manager data records.

The format of the coupling facility manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQEST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qest, qestid.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	4104	QEST	CF manager statistics
0	0	Bitstring	2	QESTID	Control block identifier
2	2	Integer	2	QESTLL	Control block length
4	4	Character	4	QESTEYEC	Control block eye catcher
8	8	Character	4096	QESTZERO	QEST part cleared on occasion
8	8	Character	64	QESTSTUC (0:63)	Array (one entry per structure)
8	8	Character	12	QESTSTR	Structure name
20	14	Integer	4	QESTSTRN	Structure number
24	18	Integer	4	QESTCSEC	Number of IXLLSTE calls
28	1C	Integer	4	QESTCMEC	Number of IXLLSTM calls
32	20	Character	8	QESTSSTC	Time spent doing IXLLSTE calls
40	28	Character	8	QESTMSTC	Time spent doing IXLLSTM calls
48	30	Integer	4	QESTRSEC	Number of IXLLSTE redrives
52	34	Integer	4	QESTRMEC	Number of IXLLSTM redrives

Offset: Dec	Offset: Hex	Type	Len	Name	Description
56	38	Integer	4	QESTSFUL	Number of structure fulls
60	3C	Integer	4	QESTMNUS	Maximum number of entries in use
64	40	Integer	4	QESTMLUS	Maximum number of elements in use
68	44	Character	4	*	Reserved
4104	1008	Character	0	*	End of control block

The data contains information for each coupling facility list structure, including the CSQ_ADMIN structure, that the queue manager could connect to during the statistics interval. The information for each structure includes the following:

- The number of and cumulative elapsed times for IXLLSTE and IXLLSTM requests.
- The number of times a request had to be retried because of a timeout.
- The number of times a 'structure full' condition occurred.

Topic manager data records

Use this topic as a reference to the format of the topic manager data records.

The format of the topic manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQTST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qtst, qtstid.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	96	QTST	Topic manager statistics
0	0	Bitstring	2	QTSTID	Control block identifier
2	2	Integer	2	QTSTLL	Control block length
4	4	Character	4	TESTEYEC	Control block eye catcher
8	8	Character	88	QTSTZERO	QTST part cleared on occasion
8	8	Integer	4	QTSTSTOT	Total subscription requests
12	0C	Integer	4	QTSTSDUR	Durable subscription requests
16	10	Integer	4	QTSTSHIG (1:3)	Subscription high water mark array (API, ADMIN, PROXY)
28	1C	Integer	4	QTSTSLOW (1:3)	Subscription low water mark array (API, ADMIN, PROXY)
40	28	Integer	4	QTSTSEXP	Subscriptions expired
44	2C	Integer	4	QTSTTMSG	Total messages put to Sub queue
48	30	Integer	4	QTSTSPHW	Single publish subscriber high water mark
52	34	Integer	4	QTSTPTOT (1:3)	Total Publication requests (API, ADMIN, PROXY)

Table 38. Topic manager statistics record (QTST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
64	40	Integer	4	QTSTPTHI	Total publish high water mark
68	44	Integer	4	QTSTPTLO	Total publish low water mark
72	48	Integer	4	QTSTPNOS	Count of publishes to no subscriber
76	4C	Integer	4	*	Reserved
80	50	Bitstring	8	QTSTETHW	Elapse time HW on publish
88	58	Bitstring	8	QTSTETTO	Elapse time total on publish

Coupling facility manager SMDS data records

Use this topic as a reference to the format of the coupling facility manager shared message data set (SMDS) data records.

The format of the coupling facility manager shared message data set (SMDS) statistics record is described in assembler macro `thlqual.SCSQMACS` (CSQDQESD) and in C header file `thlqual.SCSQC370` (CSQDSMFC).

The statistics provide information about the utilization of the owned shared message data set, I/O activity for the group of shared message data sets, and SMDS buffer utilization.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Layout of channel initiator SMF type 115 records

The layout of channel initiator statistics data (SMF type 115, subtype 231) records is described in this topic.

Self-defining section

The self-defining section for the channel initiator statistics data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure `qwsx` in the C programming language header file `thlqual.SCSQC370` (CSQDSMFC), and in assembler macro `thlqual.SCSQMACS` (CSQDQWSX).

Table 39 on page 353 shows the format of the self-defining section.

Table 39. Structure of the channel initiator statistics self-defining section

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Integer	4	QWSX0PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWSX0PSL	Length of the QWHS
6	6	Integer	2	QWSX0PSN	Number of instances of QWHS
8	8	Integer	4	QWSX0R1O	Offset from the start of the SMF record to the first channel initiator control information block (QCCT)
12	C	Integer	2	QWSX0R1L	Length of the QCCT
14	E	Integer	2	QWSX0R1N	Number of instances of QCCT

Table 39. Structure of the channel initiator statistics self-defining section (continued)

Offset: Dec	Offset : Hex	Type	Length	Name	Description
16	10	Integer	4	QWSX0R2O	Offset from the start of the SMF record to the first dispatcher task block (QCT_DSP)
20	14	Integer	2	QWSX0R2L	Length of the QCT_DSP
22	16	Integer	2	QWSX0R2N	Number of instances of QCT_DSP
24	18	Integer	4	QWSX0R3O	Offset from the start of the SMF record to the first adapter task block (QCT_ADP)
28	1C	Integer	2	QWSX0R3L	Length of the QCT_ADP
30	1E	Integer	2	QWSX0R3N	Number of instances of QCT_ADP
32	20	Integer	4	QWSX0R4O	Offset from the start of the SMF record to the first SSL task block (QCT_SSL)
36	24	Integer	2	QWSX0R4L	Length of the QCT_SSL
38	26	Integer	2	QWSX0R4N	Number of instances of QCT_SSL
40	28	Integer	4	QWSX0R5O	Offset from the start of the SMF record to the first DNS task block (QCT_DNS)
44	2C	Integer	2	QWSX0R5L	Length of the QCT_DNS
46	2E	Integer	2	QWSX0R5N	Number of instances of QCT_DNS

Typically one record contains all the data. If there are a large number of dispatchers, adapters, or SSL tasks, the data is split over more than one record.

If this happens, the count of instances of some type of tasks can be zero, and information about a group of tasks can be spread across multiple records. The channel initiator control information block (QCCT) is only present in the first record. For example the data could be split between two SMF records like this:

Table 40. Example data

Count	First record	Last record
QWHS	1	1
QCCT	1	0
QCT_DSP	50	5
QCT_ADP	0	10
QCT_SSL	0	3
QCT_DNS	0	1

This example shows that there were 55 dispatcher TCBs running during the SMF interval.

Instrumentation standard header (QWHS)

The format of the QWHS is described in structure qwhs in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 115 records:

Table 41. Key fields in the QWHS

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHSMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record. The subsystem ID in QWHSSSID, and the SMF interval start time in QWHSTIME, can be used to group multiple records for the same interval.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHS DURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

Channel initiator statistics data records

Use this topic as a reference for channel initiator statistics data records.

The format of the channel initiator statistics data record contains two parts:

- The first part is the channel initiator control information block, described in assembler macro `th1qua1.SCSQMACS(CSQDQCCT)`. For further information, see [“Channel initiator control information block”](#) on page 356.
- The second part is the channel initiator task block, described in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

The channel initiator task block contains information about the four types of task within the CHINIT. For further information, see:

- [“Dispatcher tasks”](#) on page 357
- [“Adapter tasks”](#) on page 358
- [“Domain Name Server \(DNS\) task”](#) on page 359
- [“SSL tasks”](#) on page 360

Each task includes:

- The elapsed time that the task spent processing requests in the interval (*qctel_{tm}*)
- The CPU time used by the task in the interval, which is made up of CPU used while processing requests and CPU used between requests (*qctcpt_m*)
- The total wait time of this task in the interval (*qctwt_{tm}*)
- The number of requests in the interval (*qctreq_n*)

You can use this information to see how busy the task was, and determine whether you need to add more tasks based on the analysis.

For TLS and DNS tasks, the duration of the longest request (*qctlg_{du}*, *qctls_{du}*) and the time of day when this occurred (*qctlg_{dm}*, *qctls_{dm}*) are also included.

These can be useful to identify when channel requests took a long time. For example, a DNS lookup request going to a server outside of your enterprise taking seconds rather than milliseconds.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qctelmt*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

The example accounting data in the following tasks has been formatted using IBM MQ SupportPac MP1B.

Both of the parts are also described in the C programming language header file `thlqual.SCSQC370` (CSQDSMFC). Note that the field names in C are all in lowercase, for example, *qcct*, *qct_adp*.

z/OS Channel initiator control information block

Use this topic as a reference for the channel initiator control information block.

The channel initiator control information block contains basic information for this CHINIT, including:

- CHINIT job name (*qcctjobn*)
- QSG name if it is in a queue sharing group (*qcctqsgn*)
- Peak number used of current channels (*qcctnocc*)
- Peak number used of active channels (*qcctnoac*)
- MAXCHL - maximum permitted current channels (*qcctmxcc*)
- ACTCHL - maximum permitted active channels (*qcctmxac*)
- TCPCHL - maximum permitted TCP/IP channels (*qcctmxtp*)
- LU62CHL - maximum permitted LU62 channels (*qcctmxlu*)
- **V 9.4.0** 31-bit storage used by CHINIT in the extended private region (*qcctstus*). This information is also provided by the `CSQX004I` message in the CHINIT job log.
- **V 9.4.0** 64-bit storage limit available to the CHINIT (*qcctslim*)
- **V 9.4.0** 64-bit storage used by CHINIT (*qcctstab*). This information is also provided by the `CSQX004I` message in the CHINIT job log.

The format of the channel initiator control information block is described in structure `qcct` in the C programming language header file `thlqual.SCSQC370` (CSQDSMFC), and in assembler macro `thlqual.SCSQMACS` (CSQDQCCS).

You can use this information to see if the number of active channels is approaching the configured maximum value. Note that the number of current and active channels are the values when the record was created. So, between the two intervals there might have been more than this number of channels active.

Channel information from SMF data

Here is an example of channel information from SMF data:

```
V 9.4.0
MV4A,MQ27,2023/10/02,11:53:02,VRM:934,
From 2023/10/02,11:52:52 to 2023/10/02,11:53:02, duration 10 seconds.
Peak number used of current channels..... 1
Peak number used of active channels ..... 1
MAXCHL. Max allowed current channels..... 9999
ACTCHL. Max allowed active channels..... 9999
TCPCHL. Max allowed TCP/IP channels..... 9999
LU62CHL. Max allowed LU62 channels..... 200
31-bit storage used..... 436 MB
64-bit storage limit.....16384 PB
64-bit storage used..... 187 MB
64-bit storage free.....16384 PB
```

You can monitor the storage usage and see whether the value is trending upwards. If the total used is approaching the total storage available, you might be running out of storage, and so might not be able to support many more channels.

If the numbers of active current channels are tending towards the maximum number of channels, you might need to increase the maximum number of channels.

Dispatcher tasks

This topic contains example data for the dispatcher tasks statistics, and information about how to interpret the data.

The format of the dispatcher task block is described in structure `qct_dsp` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

Example data

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	DISP	26587	0.4	0.592463	0.1	22	127
1	DISP	26963	0.3	0.588092	0.1	22	112
2	DISP	864329	2.7	2.545668	0.3	3	28
3	DISP	26875	0.4	0.590825	0.1	22	120
4	DISP	26874	0.4	0.603285	0.1	22	123
Summ	DISP	971628	0.8	4.920332	0.1	5	38

The example data shows that there were five dispatchers. A channel is associated with a dispatcher when it starts. The channel initiator tries to distribute work across all the dispatchers when allocating a channel to a dispatcher. This example shows that one dispatcher is processing more requests than other dispatchers. This is normal, as some channels might stop, so the dispatcher is processing fewer channels, and some channels can be busier than others.

- 4.9 seconds of CPU were used by the dispatchers.
- The average request used 5 microseconds of CPU and took 38 microseconds elapsed time.
- A dispatcher is used to send and receive data over a communications network, and this is not usually dependent on external events. The average elapsed time should, therefore, be close to the average CPU time used. The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests.

The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time. If the CHINIT is delayed due to lack of CPU, then the ratio of average elapsed time to average CPU time is much larger, compared to when the CHINIT is not delayed for CPU.

- The average CPU used per request depends on the message traffic. For example, bigger messages use more CPU than smaller messages.

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests : `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn`
- Average ET: `qcteltm` and `qctreqn`

Usually, the number of dispatchers should be less than, or equal to, the number of processors in the LPAR. If you have more dispatchers than processors in the LPAR they might compete for CPU resources. For more information about tuning your system, see [SupportPac MP16](#).

Channels have an affinity to a dispatcher, so you might find that some dispatchers process many more requests than another dispatcher.

You can use the ALTER QMGR CHIDISPS() command to change the number of dispatchers used. Any change comes into effect the next time the channel initiator is started.

Adapter tasks

This topic contains example data for the adapter tasks statistics, and information about how to interpret the data.

The format of the adapter task block is described in structure `qct_adp` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

Example data

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	ADAP	470297	10.2	41.290670	4.6	88	194
1	ADAP	13907	0.6	1.589428	0.2	114	365
2	ADAP	2517	0.2	0.185325	0.0	74	746
3	ADAP	1095	0.1	0.085774	0.0	78	907
4	ADAP	535	0.1	0.040743	0.0	76	947
5	ADAP	220	0.0	0.016228	0.0	74	1175
6	ADAP	82	0.0	0.005521	0.0	67	1786
7	ADAP	80	0.0	0.004248	0.0	53	1160
Summ	ADAP	488733	1.4	43.217938	0.6	88	205

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests: `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn` average
- ET: `qcteltm` and `qctreqn`

This example shows that there were eight adapter tasks.

Adapter number 0

- Processed the majority of the requests (470297 out of 488733)
- Was busy 10.2% of the interval
- Used 41.3 seconds of CPU

Overall

The average CPU per request was 88 microseconds of CPU and took 205 microseconds

The adapters process IBM MQ requests. Some of these requests might wait, for example, for log I/O during a commit, so the average Elapsed Time per request has little meaning.

The CPU time (`qctcptm`) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (`qcteltm`) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

When an IBM MQ request is made the first free adapter task is used.

- If there is at least one adapter that has been little used (less than 1%) busy, you have enough adapters.
- If at least one adapter was not used, you have enough adapters defined.
- If all the adapters were used, you might need to allocate more adapters.
- If all of the adapters were used, and they were all busy for most of the interval, you need to allocate more adapters.

You can use the ALTER QMGR CHIADAPS() command to change the number of adapters used. Any changes come into effect the next time the channel initiator is started.



Attention: If there are too many adapters acting on a small set of queues, you might get contention within the queue manager.

Related reference

[ALTER QMGR](#)



Domain Name Server (DNS) task

This topic contains example data for the DNS tasks statistics, and information about how to interpret the data.

The format of the DNS task block is described in structure `qct_dns` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCTA)`.

Example data

```
Task, Type, Requests, Busy %, CPU used, CPU %, "avg CPU", "avg ET", longest,
date,           time
0, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
Summ, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
```

The channel initiator uses a single DNS task. The example shows that the task processed 14002 requests and on average the request used 9 microseconds of CPU and took 11 microseconds of elapsed time.

The longest DNS request took 463 microseconds elapsed time, and this occurred at 12:56:33 local time.

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests : `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn`
- Average ET: `qcteltm` and `qctreqn`
- Longest: `qctlgdu`
- Longest at: `qctlgtm`

The DNS task can go out of your enterprise to look up the IP address associated with a name. If the average Elapsed time is significantly more than the average CPU time used, you might have some long requests.

If the value of the longest request time is unacceptable you should work with your network team to investigate why you are having long requests. It might be that you have an invalid name in your connections.

If the DNS task is busy for 25% of the duration, consider investigating the cause further.

The CPU time (`qctcptm`) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (`qcteltm`) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Note: There are requests to the DNS task that are not DNS lookups, so you might have the number of requests being greater than zero - but no longest request information.

This topic contains example data for the SSL tasks statistics, and information about how to interpret the data.

The format of the SSL task block is described in structure `qct_ssl` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCTA)`.

Example data

Task date,	Type,	Requests, time	Busy %,	CPU used,	CPU %,	"avg CPU",	"avg ET",	longest,	
				Seconds,		uSeconds,	uSeconds,	uSeconds,	
0,	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,	2014/03/18,
12:46:40.237697									
1,	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,	2014/03/18,
12:46:18.938022									
2,	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,	2014/03/18,
12:46:35.358145									
3,	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,	2014/03/18,
12:46:44.514045									
4,	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,	2014/03/18,
12:46:22.134123									
Summ,	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,	2014/03/18,
12:46:22.134123									

This example data shows that the average request took 364 microseconds. The longest request was for SSL task 4, took 22,438 microseconds, and occurred at 12:46:22.134123 local time.

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests : *qctreqn*
- Busy %: *qcteltm* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltm* and *qctreqn*
- Longest: *qctlsdu* longest at: *qctlstm*

A running channel is associated with an SSL task, in a similar way that a channel is associated with a dispatcher. The SSL tasks can use the cryptographic coprocessors available to the LPAR. So, the elapsed time can include time spent on a coprocessor. You should monitor the average elapsed time throughout the day. If this time increases significantly during peak periods you should work with your z/OS systems programmers, as your coprocessors might be over-used.

If the SSL tasks are busy for a significant proportion of the interval, increasing the number of SSL tasks might help. If the SSL tasks are waiting for external resources such as a coprocessor, increasing the number of SSL tasks has little effect.

You can use the ALTER QMGR SSLTASKS() command to change the number of SSL tasks used. Any changes come into effect the next time the channel initiator is started.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Related reference

[ALTER QMGR](#)

Use this topic as a reference for queue (SMF type 115, subtype 216) data records. The statistics are designed to make it easier for you to monitor usage and performance of your queue over time, and give an insight into what happened with your queue during the last SMF interval. This includes all the DISPLAY QSTATUS information and information on message flow, expiry, high and low watermarks and more.

The format of the queue statistics data record is described in assembler macro `th1qual.SCSQMACS(CSQDQQST)`.

The queue statistics record contains information on the performance of selected queues and includes the following fields:

QQSTID – Control block identifier

The identifier for the queue statistics control block; is always x' D80F '.

QQSTLL – Length of control block

The length of a queue statistics record.

QQSTEYEC – Control block eyecatcher

The eyecatcher used to make identification of the control block easier; is always ' QQST '

QQSTQNAM – Queue name

The name of the queue.

QQSTFLAG

An array of bits containing the following information about the queue:

QQSTDISP – Queue disposition

This bit identifies whether the queue is of private or shared disposition. If the bit is on, then it is a shared queue.

QQSTPART – Partial record identifier

This bit identifies whether the record is a full or partial record. If the bit is on, then it is a partial record. When this flag is set there was an issue accessing the information on the queue, for example if there is a CF structure failure.

In a shared queue partial record, the accuracy of **qqstdpth**, **qqstmage**, and **qqstuncm** cannot be guaranteed. Therefore, the fields **qqstdpth** and **qqstmage** are populated with x' 00 ' and the **qqstuncm** flag is not set.

In a private queue partial record, the accuracy of **qqstmage** cannot be guaranteed, therefore the field is populated with x' 00 '.

QQSTUNCM – Uncommitted changes pending

This bit indicates whether there are any uncommitted changes (puts and gets) pending for the queue. If the bit is on, there are uncommitted changes.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would if run at the time the SMF record was generated.

If the queue is a shared queue and QQSTPART is set, this bit is always off as there might have been an issue obtaining the correct value.

QQSTPSID – Page set ID

The page set ID where the queue is located, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a page set assigned to it, this field is set to -1 (x' FFFF ').

This value is correct at the time the SMF record was generated. It is possible that the page set changed during the SMF interval, in which case, the value reflected in the next SMF record will be the new page set.

QQSTBPID – Buffer pool ID

The buffer pool ID used by the queue, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a buffer pool assigned to it, this field is set to -1 (x' FFFF ').

This value is correct at the time the SMF record was generated. It is possible that the buffer pool changed during the SMF interval. In this case, the value reflected in the SMF record is the new buffer pool.

QQSTQSGN – QSG name

The Queue Sharing Group name that the queue manager is a member of, if it is a shared queue. If the queue is a private queue this field is blank.

QQSTCFST – CF Structure name

The coupling facility (CF) structure name the queue uses if it is a shared queue. If the queue is a private queue this field is blank.

QQSTDPH – Current depth

The depth of the queue at the time the SMF data was captured.

If the queue is a shared queue and QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

QQSTOPCT – Current open for output count

The number of handles that are currently open for output for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as OPPROCS from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTIPCT – Current open for input count

The number of handles that are currently open for input for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as IPPROCS from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTMAGE – Oldest message age

The age, in seconds, of the oldest message on the queue.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

If QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

QQSTQTST – Short term QTIME

The interval, in microseconds, between messages being put on the queue and then being destructively read. Value based on the last few messages processed. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the first value in QTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTQTLT – Long term QTIME

The interval, in microseconds, between messages being put on the queue and then being destructively read. The value is based on a larger sample of the recently processed messages. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the second value in QTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTLPUT – Last put date/time

The time, in store clock format, at which the last message was put to the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

This is the same as LPUTDATE and LPUTTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

QQSTLGET – Last get date/time

The time, in store clock format, at which the last message was retrieved from the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

A message being browsed does not count as a message being retrieved.

This is the same as LGETDATE and LGETTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTDPHI – Highest depth

The highest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPHI value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPHI is set to zero.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is higher than the current value of QQSTDPHI, then it is used as the new value of QQSTDPHI.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is higher than QQSTDPHI, if so the current queue depth is used as the new value of QQSTDPHI.

This approach means that the value of QQSTDPHI does not take into account messages put by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

QQSTDPL0 – Lowest depth

The lowest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPL0 value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPL0 is set to a special value.
- The first time during the interval the queue manager obtains the depth of the queue, QQSTDPL0 to that value.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is lower than the current value of QQSTDPL0, then it is used as the new value of QQSTDPL0.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is lower than QQSTDPL0, if so the current queue depth is used as the new value of QQSTDPL0.

This approach means that the value of QQSTDPL0 does not take into account messages got by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

QQSTPUTS – MQPUT count

The number of messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPUT1 – MQPUT1 count

The number of messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPPT – Non-persistent MQPUT count

The number of non-persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPPT – Persistent MQPUT count

The number of persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPP1 – Non-persistent MQPUT1 count

The number of non-persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPP1 – Persistent MQPUT1 count

The number of persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPUTB – MQPUT bytes

The number of bytes of message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPT1B – MQPUT1 bytes

The number of bytes of message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPPB – Non-persistent MQPUT bytes

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPPB – Persistent MQPUT bytes

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNP1B – Non-persistent MQPUT1 bytes

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTP1B – Persistent MQPUT1 bytes

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTFLPT – Failed MQPUT count

The number of MQPUT calls targeting the queue, which failed with a completion code of MQCC_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

QQSTFLP1 – Failed MQPUT1 count

The number of MQPUT1 calls targeting the queue, which failed with a completion code of MQCC_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

QQSTFPTC – Fast puts to a waiting getter count

The number of MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval.

Note: *Put to a waiting getter* is a technique whereby a message might not actually be put onto a queue if there is an application already waiting to get the message. Certain conditions must be satisfied for this to occur, in particular the message must be non-persistent and the putting and getting application must be processing the message outside syncpoint control.

If these conditions are met, then the message is transferred from the putting application's buffer into the getting application's buffer without actually touching the IBM MQ queue. This removes a lot of processing involved in putting the message on the queue and therefore leads to increased throughput and lower CPU costs.

QQSTFPTB – Fast puts to a waiting getter bytes

The number of message and properties bytes from MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval. This does not include message headers in the calculation of the size.

QQSTSTRM – Streamed message count

The number of messages that were successfully streamed from the queue during the interval. This is always zero if STREAMQ is not set for the queue.

QQSTMSMI – Minimum message size put

The minimum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTMSMA – Maximum message size put

The maximum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTMSAV – Average message size put

The average message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTGETS – Destructive MQGET count

The number of messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTNPDG – Non-persistent destructive MQGET count

The number of non-persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTPDG – Persistent destructive MQGET count

The number of persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTGETB – Destructive MQGET byte count

The number of message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTNPDB – Non-persistent destructive MQGET bytes

The number of non-persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTPDB – Persistent destructive MQGET bytes

The number of persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTBRWS – Non-destructive MQGET count

The number of messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTNPBR – Non-persistent non-destructive MQGET count

The number of non-persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTPBR – Persistent non-destructive MQGET count

The number of persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTBRWB – Non-destructive MQGET bytes

The number of message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTNPBB – Non-persistent non-destructive MQGET bytes

The number of non-persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTPBB – Persistent non-destructive MQGET bytes

The number of persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTFLGT – Failed destructive MQGET count

The number of destructive MQGET calls targeting the queue, that failed with a completion code of MQCC_FAILED, during the SMF interval.

Also included in the count is MQCC_WARNING when accompanied by a return code of MQRC_TRUNCATED_MSG_FAILED. However, not included in this count is any MQGET with a wait that receives MQRC_NO_MSG_AVAILABLE.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

QQSTNMAG – Failed destructive MQGET with MQRC_NO_MSG_AVAILABLE count

The number of destructive MQGET calls, without wait, targeting the queue, that failed with both a completion code of MQCC_FAILED and a return code of MQRC_NO_MSG_AVAILABLE, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET, without wait, attempts through the queue manager that generated the SMF record.

QQSTTMFB – Failed destructive MQGET with MQRC_TRUNCATED_MSG_FAILED count

The number of destructive MQGET calls targeting the queue, that failed with both a completion code of MQCC_WARNING and a return code of MQRC_TRUNCATED_MSG_FAILED, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

QQSTFLGW – No message available for destructive MQGET with a wait count

The number of times that there is no message available for destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

QQSTRDGW – Re-driven destructive MQGET with a wait count

The number of times that destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to get the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to get the message, QQSTFLGW is incremented by one and the MQGET goes back into a waiting state.

QQSTFLBW – No message available for non-destructive MQGET with a wait count

The number of times that there is no message available for non-destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

QQSTRDBW – Re-driven non-destructive MQGET with a wait count

The number of times that non-destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to browse the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to browse the message, QQSTFLBW is incremented by one and the MQGET goes back into a waiting state.

QQSTSAGT – Destructive MQGET with MQRC_SIGNAL_REQUEST_ACCEPTED count

The number of destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC_WARNING and a return code of MQRC_SIGNAL_REQUEST_ACCEPTED, during the SMF interval.

QQSTSABR – Non-destructive MQGET with MQRC_SIGNAL_REQUEST_ACCEPTED count

The number of non-destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC_WARNING and a return code of MQRC_SIGNAL_REQUEST_ACCEPTED, during the SMF interval.

QQSTIPHI – High watermark for IPPROC

The highest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTIPL0 – Low watermark for IPPROC

The lowest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPHI – High watermark for OPPROC

The highest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPLO – Low watermark for OPPROC

The lowest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPEN – Successful MQOPEN count

The number of times the queue was successfully opened during the SMF interval. This does not include opens performed as part of an MQPUT1 call.

For shared queues, the count only includes when the queue is opened through the queue manager that generated the SMF record.

QQSTCLOS – MQCLOSE count

The number of times the queue was successfully closed using MQCLOSE, during the SMF interval.

For shared queues, the count only includes when the queue is closed through the queue manager that generated the SMF record.

QQSTINQR – MQINQ count

The number of MQINQ calls that completed with a completion code of MQCC_OK or MQCC_WARNING, during the SMF interval.

QQSTSET – MQSET count

The number of MQSET calls that completed with a completion code of MQCC_OK during the SMF interval.

QQSTEXPR – Expired messages count

The number of expired messages cleared from the queue during the SMF interval.

This includes messages expired by an application issuing an MQGET, by the expired message scanning task or by a REFRESH QMGR TYPE(EXPIRY) command.

QQSTRBPT – Rolled back MQPUT counts

The number of messages that were put to the queue, that have been rolled back off the queue, during the SMF interval.

QQSTRBGT – Rolled back MQGET counts

The number of messages destructively read from the queue, that have been rolled back onto the queue, during the SMF interval.

 **Interpreting IBM MQ for z/OS accounting data**

IBM MQ for z/OS accounting data is written as SMF type 116 records. Use this topic as a reference to the different types of accounting data records.

IBM MQ accounting information can be collected for the following subtypes:

0

Message manager accounting records (how much processor time was spent processing IBM MQ API calls and the number of MQPUT and MQGET calls). This information is produced when a named task disconnects from IBM MQ, and so the information contained within the record might cover many hours.

1

Accounting data for each task, at thread and queue level.

2

Additional queue-level accounting data (if the task used more queues than could fit in the subtype 1 record).

10

Accounting data for channels.

Note: Accounting information for specific channels can be enabled or suppressed by the **STATCHL** channel attribute, and the **STATACLS** queue manager attribute.

Note that:

- Subtype 0 records are produced with accounting trace class 1.
- Subtype 1 and 2 records are produced with accounting trace class 3.
- Subtype 10 records are produced with accounting trace class 4.

Layout of an SMF type 116 record

Use this topic as a reference to the format of an SMF type record.

The standard layout for SMF records involves three parts:

SMF header

Provides format, identification, and time and date information about the record itself.

Self-defining section

Defines the location and size of the individual data records within the SMF record.

Data records

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

The SMF header

Table 42 on page 369 shows the format of SMF record header (SM116).

Offset : Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM116	SMF record header.	
0	0	Integer	2	SM116LEN	SMF record length.	01A4
2	2		2		Reserved.	
4	4	Integer	1	SM116FLG	System indicator.	5E
5	5	Integer	1	SM116RTY	Record type. The SMF record type, for IBM MQ accounting records this is always 116 (X'74').	74
6	6	Integer	4	SM116TME	Time when SMF moved record.	00356124
10	A	Integer	4	SM116DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM116SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM116SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM116STF	Record subtype.	0000
24	18	Character	3	SM116REL	IBM MQ version.	F9F3F0 (930)
27	1B		1		Reserved.	
28	1C	Character	0	SM116END	End of SMF header and start of self-defining section.	

Note: The (hexadecimal) values in the right-hand column relate to [Figure 22 on page 371](#).

Self-defining sections

A self-defining section of an SMF record tells you where to find an accounting record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at a fixed offset from the start of the SMF record.

Each self-defining section points to accounting related data. [Table 43 on page 370](#) summarizes the offsets from the start of the SMF record header.

Table 43. Offsets to self-defining sections

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
All	Common header	28	X'1C'	“Common IBM MQ SMF header” on page 372
0	Message manager	44	X'2C'	“Message manager data records” on page 374
1	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 375
1	Thread-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 375
1	Queue-level accounting	52	X'34'	“Thread-level and queue-level data records” on page 375 . This section is present only if the WTASWQCT field in the task-related information (WTAS) structure is non-zero.
2	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 375
2	Queue-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 375
10	Channel accounting			“Channel accounting data records” on page 378

Note: Other self-defining sections refer to data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssl111nnnn
```

where:

SSSSSSSS

Fullword containing the offset from start of the SMF record.

LLLL

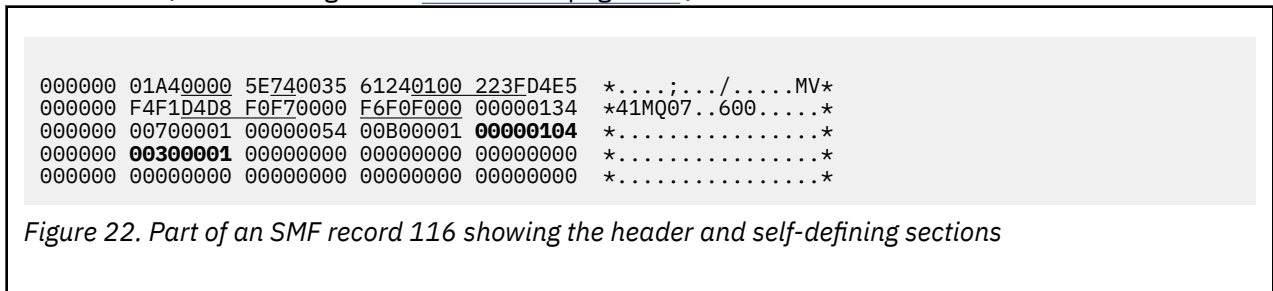
Halfword giving the length of this data record.

nnnn

Halfword giving the number of data records in this SMF record.

Figure 22 on page 371 shows an example of part of an SMF type 116 record. The numbers in the left-hand column represent the offset, in hexadecimal, from the start of the record. Each line corresponds to sixteen bytes of data, where each byte is two hexadecimal characters, for example 0C. The characters in the right-hand column represent the printable characters for each byte. Non-printable characters are shown by a period (.) character.

In this example, alternate fields in the SMF header are underlined to help you to see them; refer to Table 42 on page 369 to identify them. The self defining section for one of the message manager accounting data records (at the offset given in Table 43 on page 370) is shown in **bold**.



The self-defining section for the type of message manager accounting data is located at offset X'2C' from the start of the SMF record and contains this information:

- The offset of the message manager accounting data is located X'00000104' bytes from the start of the SMF record.
- This message manager record is X'0030' bytes long.
- There is one record (X'0001').

Note: Always use offsets in the self-defining sections to locate the accounting records.

Processing type 116 SMF records

Use this topic as a reference to the format of the processing type accounting record.

Any accounting data you collect from SMF must be processed to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the value of the following fields:

- SM116RTY, the SMF record number = X'74' (116)
- SM116STF, the record subtype, must be 0000, 0001, 0002, or 0010

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility” on page 328](#).

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

You need to update the SMFIN DD card with the name of the SMF data set. Use the z/OS command '/D SMF' to show the name of the data set, and you need to update the DUMPOUT DD card with the name for the output data set.

You also need to specify the START and END times that you require.

The following sample JCL extracts SMF records from SMF data sets and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=
//SMFIN DD DSN=xxxxxx.MANA,DISP=SHR
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
INDD(SMFIN,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

The following sample JCL extracts SMF records from the SMF log stream named in LSNAME and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=*
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
LSNAME(IFASMF.MQ,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

Common IBM MQ SMF header

Use this topic as a reference to the common IBM MQ SMF header type accounting record.

The format of this record is described in Table 44 on page 372 and in assembler macros thlqual.SCSQMACS(CSQDQWHS) and thlqual.SCSQMACS(CSQDQWHC), and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qwhs, qwhsnsda.

The QWHS data includes the subsystem name. For subtype 1 records, it also shows whether there are queue-level accounting records present. If the QWHSNSDA field is 3 or less, there are not, and the corresponding self-defining section (at offset X'34') is not set.

The QWHC data gives you information about the user (for example, the user ID (QWHCAID) and the type of application (QWHCATYP)). The QWHC section is completed only for subtype 0 records. The equivalent information is present in the thread identification record for subtype 1 and 2 records.

<i>Table 44. Structure of the common IBM MQ SMF header record QWHS</i>					
Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Structure	128	QWHS	
0	0		6		Reserved
6	6	Character	1	QWHSNSDA	Number of self defining sections in the SMF records
7	7		5		Reserved
12	C	Character	4	QWHSSSID	Subsystem name
16	10		24		Reserved
40	28	Character	8	QWHCAID	User ID associated with the z/OS job
48	30	Character	12	QWHCCV	Thread cross-reference
60	3C	Character	8	QWHCCN	Connection name
68	44		8		Reserved

Table 44. Structure of the common IBM MQ SMF header record QWHS (continued)

Offset: Dec	Offset : Hex	Type	Length	Name	Description
76	4C	Character	8	QWHCOPID	User ID associated with the transaction
84	54	Integer	4	QWHCATYP	Type of connecting system (1=CICS, 2=Batch or TSO, 3=IMS control region, 4=IMS MPP or BMP, 5=Command server, 6=Channel initiator, 7=RRS Batch)
88	58	Character	22	QWHCTOKN	Accounting token set to the z/OS accounting information for the user
110	6E	Character	16	QWHCNID	Network identifier
126	7E		2		Reserved

Combining CICS and IBM MQ performance data

Use this topic as a reference to the combination of IBM MQ and CICS performance data.

The common IBM MQ SMF header type accounting record section, QWHCTOKN, is used to correlate CICS type 110 SMF records with IBM MQ type 116 SMF records.

CICS generates an LU6.2 unit-of-work token, for each CICS task. The token is used to generate an accounting token that is written to QWHCTOKN in the correlation header of subtype zero records.

Details are also written to the WTIDACCT section in subtype 1 and 2 records. The accounting token enables correlation between CICS and IBM MQ performance data for a transaction.

Thread cross-reference data

Use this topic as a reference to the format of the thread cross-reference type accounting record.

The interpretation of the data in the thread cross-reference (QWHCCV) field varies. This depends on what the data relates to:

- CICS connections (QWHCATYP=1) - see [Table 45 on page 373](#)
- IMS connections (QWHCATYP=3 or 4) - see [Table 46 on page 373](#)
- Batch connections (QWHCATYP=2 or 7) - this field consists of binary zeros
- Others - no meaningful data

Table 45. Structure of the thread cross-reference for a CICS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	CICS thread number.
52	34	Character	4	CICS transaction name.
56	38	Integer	4	CICS task number.

Some entries contain blank characters. These apply to the task, rather than to a specific transaction.

Table 46. Structure of the thread cross-reference for an IMS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	IMS partition specification table (PST) region identifier.

Table 46. Structure of the thread cross-reference for an IMS system (continued)

Offset: Dec	Offset: Hex	Type	Length	Description
52	34	Character	8	IMS program specification block (PSB) name.

z/OS Message manager data records

Use this topic as a reference to the format of the message manager accounting records.

The message manager is the component of IBM MQ that processes all API requests. The format of the message manager accounting records is described in assembler macro thlqual.SCSQMACS(CSQDQMAC).

The QMAC data gives you information about the processor time spent processing IBM MQ calls, and counts of the number of MQPUT and MQGET requests for messages of different sizes.

Note: A single IMS application might write two SMF records. In this case, add the figures from both records to provide the correct totals for the IMS application.

Records containing zero processor time

Records are sometimes produced that contain zero processor time in the QMACCPUT field. These records occur when long running tasks identified to IBM MQ either terminate or are prompted to output accounting records by accounting trace being stopped. Such tasks exist in the CICS adapter and in the channel initiator (for distributed queuing). The number of these tasks with zero processor time depends upon how much activity there has been in the system:

- For the CICS adapter, this can result in up to nine records with zero processor time.
- For the channel initiator, the number of records with zero processor time can be up to the sum of `Adapters + Dispatchers + 6`, as defined in the queue manager attributes.

These records reflect the amount of work done under the task, and can be ignored.

z/OS Sample subtype zero accounting record

Use this topic as a reference to the format of the subtype zero accounting records.

Figure 23 on page 374 shows a type 116, subtype zero SMF record. In this figure, the SMF record header and the QMAC accounting data record are underlined. The self-defining sections are in bold.

```

000000 01A40000 5E740035 61240100 223FD4E5 *...;.../....MV*
000010 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000020 00700001 00000054 00B00001 00000104 *.....*
000030 00300001 00000000 00000000 00000000 *.....*
000040 00000000 00000000 00000000 00000000 *.....*
000050 00000000 B478AB43 9C6C2280 B478AB47 *.....%.....*
000060 9DB47E02 00000000 04C0F631 00000001 *.=.....6.....*
000070 9880E72D 00000000 014D9540 00000000 *..X.....(. ....*
000080 08480C80 00000010 40404040 40404040 *..... *
000090 00000000 00000000 00000051 00000000 *.....*
0000A0 00000000 00000000 00000000 00000000 *.....*
0000B0 00000000 00000000 00000000 00000000 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 00000000 *.....*
000100 00000000 D4140030 D8D4C1C3 00000000 *...M...QMAC...*
000110 689C738D 00000050 00000000 00000050 *.....&.....&*
000120 0000000A 00000000 00000000 00000000 *.....*
000130 00000000 0024011A 00030710 02DAACF0 *......0*

```

Figure 23. Example SMF type 116, subtype zero record

Thread-level and queue-level data records

Use this topic as a reference to the format of the thread-level and queue-level accounting records.

Thread level accounting records are collected for each task using IBM MQ. In addition, queue-level accounting records are gathered about each queue that the task opens. A queue-level accounting record is written for each queue that the task has used since the thread-level accounting record was last written.

If the task uses a queue that is configured with a streaming queue, there is no queue-level accounting record for the streaming queue. Instead, the accounting record for the original queue accumulates data for the data points that would have been associated with the streaming queue.

The only exception to this is that the PUTN/PUT1N value shows the number of MQPUT/MQPUT1 requests made by the application, and excludes the extra MQPUT requests made to the streaming queue.

So, for example, if an application issues a single MQPUT request, the:

- PUTN value is 1
- Elapsed time (PUTET) and CPU time (PUTCT) for the MQPUT include the time taken to put to both the primary and streaming queue
- Number of page set requests (PUTPSN) includes those for both the primary and secondary queue, and so on

For each task, data is written to SMF when the task finishes.

From IBM MQ 9.3.0 onwards, for long running tasks, data is also written at the interval specified by either the ACCTIME, or STATIME, parameter of the CSQ6SYSP system parameter macro, or by the system SMF statistics broadcast, provided that the task was running the previous time data was gathered.

Thread-level and queue-level accounting records are produced if you specify class 3 when you start the accounting trace. For example, use the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
```

The thread level accounting information is written to an SMF type 116, subtype 1 record, and is followed by queue-level records. If the task opened many queues, further queue information is written to one or more SMF type 116 subtype 2 records. A thread identification control block is included in each subtype 1 and 2 record to enable you to relate each record to the correct task. Typically, the maximum number of queue-level records in each SMF record is about 45.

The format of the thread-level accounting record is described in assembler macro `th1qua1.SCSQMACS(CSQDWTAS)`. The format of the queue-level accounting record is described in assembler macro `th1qua1.SCSQMACS(CSQDWQ)`. The format of the thread identification record is described in assembler macro `th1qua1.SCSQMACS(CSQDWTID)`. All these records are also described in C header file `th1qua1.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `wtas`, `wtasshex`.

Meaning of the channel names

Use this topic as a reference to the meaning of channel names.

The channel name in the WTID is constructed as shown in the following example. In this example a sender channel exists from queue manager QM1 to queue manager QM2.

The meaning of channel names are described in the following table.

<i>Table 47. Meaning of channel names</i>		
Field name	Meaning	Example
For queue manager QM1 the sender channel has the following fields set:		
WTIDCCN	The job name	QM1CHIN
WTIDCHL	The channel name	QM1.QM2

Table 47. Meaning of channel names (continued)

Field name	Meaning	Example
WTIDCHLC	This is defined in the CONNAME of the channel	WINMVS2B(2162)
For queue manager QM2 the receiver channel has the following fields set:		
WTIDCCN	The job name	QM2CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	Where the channel came from	9.20.101.14

z/OS Sample subtype 1 and subtype 2 records

Use this topic as a reference to the format of the subtype 1 and subtype 2 accounting records.

Figure 24 on page 376 and Figure 25 on page 376 show examples of SMF type 116, subtype 1 and subtype 2 records. These two accounting records were created for a batch job that opened 80 queues. Because many queues were opened, a subtype 2 record was required to contain all the information produced.

```

000000 703C0000 5E74002D 983B0100 229FD4E5 *...;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 00006FCC *41MQ07..600...?.*
000020 00700001 0000003C 00D00001 0000010C *.....}.....*
000030 02C00001 000003CC 02400030 F70000D0 *.{.....7..}*
000040 E6E3C9C4 00000000 00000000 00000040 *WTID.....*
.
.
000100 00000000 00000000 7F4A4BB8 F70102C0 *....."...7..}*
000110 E6E3C1E2 B4802373 0BF07885 7F4AE718 *WTAS.....0..".X.*

```

Figure 24. Example SMF type 116, subtype 1 record

The first self-defining section starts at X'24' and is **bold** in the example; X'0000003C' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'0000010C' is the offset to the WTAS data record, X'02C0' is the length of the WTAS record, and X'0001' is the number of WTAS records.

The third self-defining section starts at X'34' and is **bold** in the example; X'000003CC' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0030' is the number of WQST records.

Figure 25 on page 376 shows an example of an SMF type 116, subtype 2 record.

```

000000 49740000 5E74002D 983B0100 229FD4E5 *...;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 00004904 *41MQ07..600....*
000020 00700001 00000034 00D00001 00000104 *.....}.....*
000030 02400020 F70000D0 E6E3C9C4 00000002 *. .7..}WTID....*
.
.
000100 7F4A4BB8 F7020240 E6D8E2E3 00000001 *"...7.. WQST....*

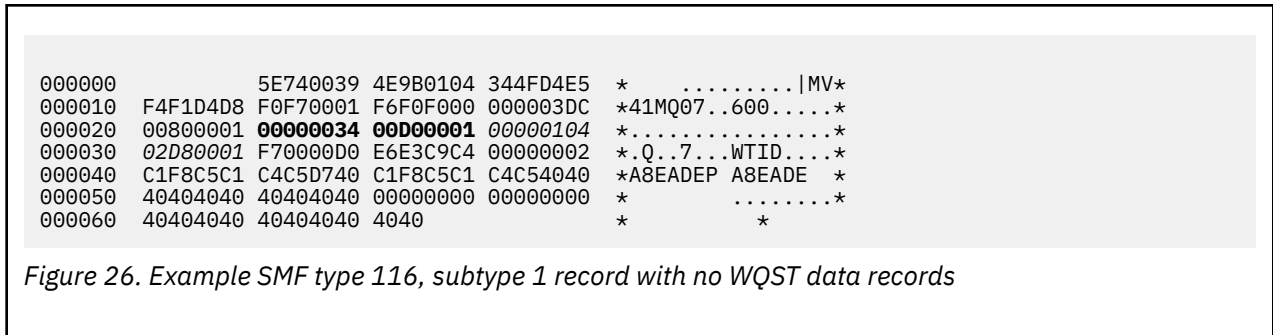
```

Figure 25. Example SMF type 116, subtype 2 record

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'00000104' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0020' is the number of WQST records.

Figure 26 on page 377 shows an example of an SMF type 116, subtype 1 record where no queues have been opened and there are consequently no self-defining sections for WQST records..



The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'0000010C' is the offset to the WTAS data record, X'02D8' is the length of the WTAS record, and X'0001' is the number of WTAS records.

There is no self-defining section describing a WQST data record, equivalent to the third self-defining section in Figure 24 on page 376.

Layout of channel initiator SMF type 116 records

The layout of channel accounting data (SMF type 116, subtype 10) records is described in this topic.

Self-defining section

The self-defining section for the channel accounting data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure qws5 in the C programming language header file th1qua1.SCSQC370 (CSQDSMFC), and in assembler macro th1qua1.SCSQMACS (CSQDQWS5).

Table 48 on page 377 shows the format of the self-defining section.

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Integer	4	QWS50PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWS50PSL	Length of the QWHS
6	6	Integer	2	QWS50PSN	Number of instances of QWHS
8	8	Integer	4	QWS50R1O	Offset from the start of the SMF record to the first channel accounting data record (QCST)
12	C	Integer	2	QWS50R1L	Length of the QCST
14	E	Integer	2	QWS50R1N	Number of instances of QCST

Instrumentation standard header (QWHS)

The format of the QWHS is described in structure `qwhs` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 116 records:

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHS SMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHS DURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

Channel accounting data records

Use this topic as a reference for channel accounting data records.

The format of the channel accounting data record is described in assembler macro `thlqual.SCSQMACS(CSQDQCST)`. The format is also described in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, `qcst`.

The channel accounting data gives you information about the status and statistics of each channel instance, including:

- Average network time (`qcstntav`)
- Average time on exit (`qcstetav`)
- Channel batch data limit (`qcstc bdl`)
- Channel batch interval (`qcstc bit`)
- Channel batch size (`qcstc bsz`)
- Channel dispatcher number (`qcstd spn`)
- Channel disposition (`qcstc hdp`)
- Channel name (`qcstc hnm`)
- Channel state (`qcstc hst`)
- Channel started time (`qcstc strt`)
- Channel status collected time (`qcstc ltm`)
- Channel stopped time (`qcstc ldt`)
- Channel type (`qcstc hty`)
- Common name (CN) from SSLCERTI (`qcstc lcn`)
- Compression rate (`qcstc pra`)
- Connection name (`qcstc nnm`)
- Current shared conversations (`qcstc scv`)

- DNS resolution time (*qcstdnrt*)
- Effective value of STATCHL parameter (*qcststcl*)
- Last message time (*qcstlmst*)
- Maximum network time (*qcstntmx*)
- Maximum time on exit (*qcstetmx*)
- Minimum network time (*qcstntmn*)
- Minimum time on exit (*qcstetmn*)
- Name of the remote queue manager or application (*qcstrqmn*)
- Number of batches (*qcstbatc*)
- Number of bytes for message data (*qcstnbyt*)
- Number of bytes for persistent message data (*qcstnpby*)
- Number of bytes received for both message data and control information (*qcstbyrc*)
- Number of bytes sent for both message data and control information (*qcstbyst*)
- Number of full batches (*qcstfuba*)
- Number of messages, or number of MQI calls (*qcstnmsg*)
- Number of persistent messages (*qcstnmsg*)
- Number of put retries (*qcstptrc*)
- Number of transmission queue becoming empty (*qcstqetc*)
- Number of transmission buffers received (**qcstbfrc**)
- Number of transmission buffers sent (*qcstbfst*)
- Serial number from SSLPEER (*qcstslsn*)
- SSL CipherSpec (zero means TLS not used) (*qcstslcs*)
- The date and time of maximum network time (*qcstntdt*)
- The date and time of maximum time on exit (*qcstetdt*)

Note, that for the channel accounting field *qcstetmn* (Minimum time on exit) and *qcstntmn* (Minimum network time) these two fields will be initialized to the hexadecimal value of 8FFFFFFF when unused.

You can use this information to see the throughput of a channel, if the actual batches are approaching the limit, the latency of the network, information about the remote end, performance of user exit, and so on.

Here is an example of the channel accounting data which has been formatted with IBM MQ SupportPac MP1B.

The fields available are based on the display channel status command (DIS CHS) and channel statistics by IBM MQ on platforms except z/OS, with some additional fields.

The data and time of the start and end of the record in local time, and the duration

```
SMF interval start      2014/03/26,02:30:00
SMF interval end       2014/03/26,02:45:00
SMF interval duration   899.997759 seconds
```

Information about the channel

```
Connection name      9.20.4.159
Channel disp         PRIVATE
Channel type         RECEIVER
Channel status       CLOSING
Channel STATCHL      HIGH
```

```
Start date & time     2014/03/26,02:44:58
Channel status collect time 2014/03/26,02:45:00
Last status changed   1900/01/01,00:00:00
Last msg time         2014/03/26,02:44:59
```

```

Batch size                50
Messages/batch           3.3
Number of messages       1,102
Number of persistent messages 1,102
Number of batches        335
Number of full batches    0
Number of partial batches 335
Buffers sent             337
Buffers received         1,272
Message data             5,038,344  4 MB
Persistent message data  5,038,344  4 MB
Non persistent message data 0  0 B
Total bytes sent         9,852  9 KB
Total bytes received     5,043,520  4 MB
Bytes received/Batch     15,055  14 KB
Bytes sent/Batch         29  29 B
Batches/Second           1
Bytes received/message   4,576  4 KB
Bytes sent/message       8  8 B
Bytes received/second    28,019  27 KB/sec
Bytes sent/second        54  54 B/sec
Compression rate         0

```

```

The name of the queue manager at the remote end of the connection
Remote qmgr/app          MQPH
Put retry count          0

```

Ajuste de la red de IBM MQ

Utilice los consejos de ajuste de esta sección como ayuda para mejorar el rendimiento de la red del gestor de colas.

Ajuste canales de conexión cliente y servidor

El valor predeterminado para **SHARECNV** es 10, que permite hasta 10 conversaciones de cliente para cada instancia de canal. Sin embargo, el uso de un número diferente de conversaciones compartidas puede ser mejor para el rendimiento. Si no necesita conversaciones compartidas o está utilizando un servidor distribuido, establezca **SHARECNV** en 1. Si tiene aplicaciones cliente existentes que no se ejecutan correctamente cuando establece **SHARECNV** en 1 o superior, establezca **SHARECNV** en 0.

Acerca de esta tarea

En algunas configuraciones, el uso de conversaciones compartidas aporta ventajas significativas. Sin embargo, en servidores distribuidos, el proceso de mensajes en canales que utilizan la configuración predeterminada de 10 conversaciones compartidas es un promedio del 15% más lento que en canales que no utilizan conversaciones compartidas. En una instancia de canal de MQI que está compartiendo conversaciones, todas las conversaciones en un socket las recibe la misma hebra. Si las conversaciones que comparten un socket están todas ocupadas, las hebras de la conversación compiten entre sí para utilizar la hebra receptora. La competición causa retrasos, y en esta situación conviene utilizar un número más pequeño de conversaciones compartidas.

Utilice el parámetro **SHARECNV** para especificar el número máximo de conversaciones que deben compartirse a través de una instancia de canal de cliente TCP/IP determinada. Para obtener detalles de todos los valores posibles, consulte [Cliente soportado IBM MQ : comportamiento predeterminado de los canales de conexión de cliente y de conexión de servidor](#).

Si establece **SHARECNV** en 1 o superior, habilita las siguientes mejoras de rendimiento:

- Pulsaciones bidireccionales
- Detención-inmovilización de administrador
- Lectura anticipada
- Consumo asíncrono por aplicaciones cliente

Si no necesita conversaciones compartidas, estos dos valores ofrecen el mejor rendimiento:

- **SHARECNV(1)**.

- SHARECNV(0).

Notas:

- Si el valor **SHARECNV** de la conexión con el cliente no coincide con el valor **SHARECNV** de la conexión con el servidor, se utiliza el valor más bajo.
- Cuando las aplicaciones se enlazan o compilan con una biblioteca no reentrante, el valor CURSHCNV(0) se negocia incluso si se establece un valor superior en CLNTCONN y SVRCONN.

Para optimizar el rendimiento para una instancia de canal determinada, realice cualquiera de los pasos siguientes.

Procedimiento

- Supervise los canales que utilizan el valor predeterminado de SHARECNV de 10.

El valor predeterminado de SHARECNV(10) funciona bien en muchos escenarios, pero es posible que no sea el valor óptimo para una instancia de canal determinada. Por ejemplo, para los servidores distribuidos, el proceso de mensajes en los canales que utilizan este valor es, de media, un 15% más lento que en los canales que no utilizan conversaciones compartidas.

Para asegurarse de que el valor predeterminado es adecuado para una instancia de canal determinada, supervise el rendimiento del canal con este valor.

- Establezca un valor SHARECNV de 2 o más.

Puede establecer SHARECNV(2) en SHARECNV(99999999). Para asegurarse de que el valor que elija es adecuado para una instancia de canal determinada, supervise el rendimiento del canal con el nuevo valor.

- Establezca un valor SHARECNV de 1.

Si no necesita conversaciones compartidas, utilice este valor siempre que sea posible. Elimina la contención para utilizar la hebra receptora, y las aplicaciones cliente pueden aprovechar las mejoras de rendimiento descritas en la sección "sobre esta tarea".

Con este valor, el rendimiento del servidor distribuido mejora significativamente. Las mejoras de rendimiento se aplican a las aplicaciones cliente que emiten llamadas de espera de obtención síncrona no de lectura anticipada; por ejemplo, llamadas de espera MQGET de cliente C. Cuando estas aplicaciones cliente se conectan, el servidor distribuido utiliza menos hebras y menos memoria y se aumenta el rendimiento.

Si un servidor tiene clientes conectados que comparten conversaciones a través de un socket, y reduce el valor de conversaciones compartidas de SHARECNV(10) a SHARECNV(1), verá los siguientes efectos:

- Se incrementa el uso de socket en el servidor.
- Se incrementan las instancias de canal en el servidor.

En este caso, también puede optar por aumentar los valores para **MaxChannels** y **MaxActiveChannels**.

Nota: También puede establecer la opción MQCONNX, MQCNO_NO_CONV_SHARING y conectar la aplicación a un canal con **SHARECNV** establecido en un valor mayor que 1. El resultado es el mismo que conectar la aplicación a un canal con **SHARECNV** establecido en 1.

- Establezca un valor de SHARECNV de 0.

La instancia de canal se comporta exactamente como si fuera un canal de conexión de servidor o cliente de IBM WebSphere MQ 6.0. No obtiene conversaciones compartidas ni las mejoras de rendimiento que están disponibles cuando establece **SHARECNV** en 1 o superior. Utilizar un valor de 0 sólo si tiene aplicaciones cliente existentes que no se ejecutan correctamente cuando se establece **SHARECNV** en 1 o superior.

Conceptos relacionados

Cliente de IBM MQ soportado: comportamiento predeterminado de los canales de conexión de cliente y de conexión de servidor

Ajuste de redes de publicación/suscripción distribuidas

Utilice los consejos de ajuste de esta sección para ayudar a mejorar el rendimiento de sus clústeres y jerarquías de publicación/suscripción distribuida de IBM MQ.

Conceptos relacionados

“Supervisión de clústeres” en la página 318

Dentro de un clúster puede supervisar mensajes de aplicaciones, mensajes de control y registros. Existen consideraciones de supervisión especiales cuando la carga del clúster se equilibra entre dos o más instancias de una cola.

Rendimiento de los clústeres de publicación/suscripción de direccionamiento directo

En los clústeres de publicación/suscripción de direccionamiento directo, la información, tal como los temas en clúster y las suscripciones de proxy, se transfiere a todos los miembros del clúster, independientemente de si todos los gestores de colas del clúster participan de forma activa en la mensajería de publicación/suscripción. Este proceso puede crear una importante carga adicional sobre el sistema. Para disminuir el efecto que la gestión de clústeres puede tener en el rendimiento, realice las actualizaciones en los periodos de tiempo de poca actividad, defina un subconjunto de gestores de colas mucho más pequeño para la publicación/suscripción y convierta dicho subconjunto en un clúster "solapante" o pase a utilizar el direccionamiento de host de tema.

En un gestor de colas de un clúster de publicación/suscripción, la carga de trabajo procede de dos orígenes:

- El manejo directo de los mensajes para los programas de aplicación.
- El manejo de los mensajes y canales necesarios para gestionar el clúster.

En clúster punto a punto típico, la carga de trabajo del sistema del clúster se limita prácticamente a la información que solicitan de forma explícita los miembros del clúster, según sea necesaria. Por lo tanto, en cualquier clúster que no sea un clúster punto a punto de gran tamaño, por ejemplo, un clúster con miles de gestores de colas, por lo general, puede descartar el efecto que la gestión del clúster puede tener sobre el rendimiento. Sin embargo, en un clúster de publicación/suscripción de direccionamiento directo, se transfiere a todos los miembros la información, por ejemplo, los temas del clúster, los miembros del gestor de colas y las suscripciones de proxy, independientemente de si todos los gestores de colas participan activamente en la mensajería de publicación/suscripción. Esto puede crear una importante carga adicional sobre el sistema. Por lo tanto, debe considerar el efecto que la gestión del clúster tiene sobre el rendimiento del gestor de colas, tanto en lo que respecta al tiempo como al tamaño.

Características de rendimiento de los clústeres de direccionamiento directo

Compare un clúster punto a punto con un clúster de publicación/suscripción en lo relativo las tareas de gestión principales.

En primer lugar, un clúster punto a punto:

1. Cuando se define una nueva cola de clúster, la información de destino se envía a los gestores de colas de repositorio completo, y únicamente se envía a los otros miembros del clúster cuando éstos hacen referencia, por primera vez, a una cola del clúster (por ejemplo, cuando una aplicación intenta abrirla). Luego el gestor de colas almacena en memoria caché localmente esta información para eliminar la necesidad de recuperar la información cada vez que se accede a la cola.
2. Añadir un gestor de colas a un clúster no afecta directamente a la carga en otros gestores de colas. Se envía información sobre el nuevo gestor de colas a los repositorios completos, pero los canales al

nuevo gestor de colas desde otros gestores de colas del clúster sólo se crean y se inician cuando el tráfico empieza a fluir hacia o desde el nuevo gestor de colas.

En resumen, la carga en un gestor de colas de un clúster punto a punto está relacionada con el tráfico de mensajes que maneja para los programas de aplicación y no está directamente relacionada con el tamaño del clúster.

En segundo lugar, un clúster de publicación/suscripción de direccionamiento directo:

1. Cuando se define un nuevo tema de clúster, la información se transfiere a los gestores de colas de repositorio completo y, desde allí, directamente a todos los miembros del clúster, lo que hace que se inicien los canales en cada miembro del clúster desde los repositorios completos si todavía no se han iniciado. Si este es el primer tema del clúster directo, a cada miembro del gestor de colas se le envía información acerca de los otros miembros del gestor de colas del clúster.
2. Cuando se crea una suscripción a un tema de clúster en una nueva serie de tema, la información se envía directamente desde ese gestor de colas a los demás miembros del clúster inmediatamente, lo que hace que los canales se inicien en cada miembro del clúster desde ese gestor de colas, si no se han iniciado previamente.
3. Cuando un gestor de colas nuevo se une a un clúster existente, los gestores de colas de repositorio completo transfieren a este nuevo gestor de colas la información acerca de todos los temas del clúster (y de todos los miembros del gestor de colas si se ha definido un tema de clúster directo). A continuación, el gestor de colas nuevo sincroniza la información de todas las suscripciones a los temas de clúster incluidos en el clúster con todos los miembros del clúster.

En resumen, la carga de la gestión del clúster en cualquier gestor de colas de un clúster de publicación/suscripción de direccionamiento directo aumenta según el número de gestores de colas, los temas de clúster y los cambios de las suscripciones a series de temas diferentes dentro del clúster, independientemente del uso local de estos temas de clúster en cada gestor de colas.

En un clúster de gran tamaño, o en uno en el que el índice de cambios de suscripciones sea elevado, este nivel de gestión del clúster puede ser una importante sobrecarga para todos los gestores de colas.

Reducción del efecto sobre el rendimiento de la publicación/suscripción de direccionamiento directo

Para disminuir el efecto que puede tener la gestión de un clúster de publicación/suscripción de direccionamiento directo sobre el rendimiento, tenga en cuenta las tres opciones siguientes:

- Aplique actualizaciones de clústeres, temas y suscripciones en horas de menor actividad del día.
- Defina un subconjunto mucho más pequeño de gestores de colas que participan en la publicación/suscripción y convierta este subconjunto en un clúster "solapado". Este clúster será el clúster donde se definan los temas de clúster. Aunque algunos gestores de colas están ahora en dos clústeres, se reduce el efecto general de publicación/suscripción:
 - El tamaño del clúster de publicación/suscripción es menor.
 - A los gestores de colas que no están en el clúster de publicación/suscripción les afecta en menor medida el tráfico de gestión de clústeres.

Si las opciones anteriores no resuelven de forma adecuada sus problemas de rendimiento, en su lugar, puede utilizar un clúster de publicación/suscripción *de direccionamiento de host de tema*. Para obtener una comparación detallada del direccionamiento directo y el direccionamiento de host de tema en los clústeres de publicación/suscripción [Diseño de clústeres de publicación/suscripción](#).

Conceptos relacionados

Rendimiento de los clústeres de publicación/suscripción de direccionamiento de host de tema
Un clúster de publicación/suscripción de direccionamiento de host de tema le permite controlar exactamente qué gestores de colas alojan cada tema. Estos hosts de tema se convierten en los gestores de colas de *direccionamiento* para esa rama del árbol de temas. Además, los gestores de colas sin suscripciones o publicadores no necesitan conectarse a los hosts de tema. Esta configuración puede

reducir significativamente el número de conexiones entre los gestores de colas del clúster y la cantidad de información que se pasa entre los gestores de colas.

Equilibrio entre productos y consumidores en las redes de publicación/suscripción

Un concepto importante en el rendimiento de la mensajería asíncrona es el *equilibrio*. A menos que los consumidores de mensajes estén equilibrados con los productores de mensajes, existe el riesgo de que se acumule el retraso de los mensajes no consumidos y afecte gravemente al rendimiento de varias aplicaciones.

Rendimiento de las suscripciones en las redes de publicación/suscripción

La publicación/suscripción en IBM MQ funciona de modo que se propaga el conocimiento acerca del lugar donde se han creado las suscripciones a diferentes series de temas en la red del gestor de colas. Esto permite que el gestor de colas en el que se publica un mensaje identifique los otros gestores de colas que requieren una copia del mensaje publicado, para que coincida con sus suscripciones.

Rendimiento de los clústeres de publicación/suscripción de direccionamiento de host de tema

Un clúster de publicación/suscripción de direccionamiento de host de tema le permite controlar exactamente qué gestores de colas alojan cada tema. Estos hosts de tema se convierten en los gestores de colas de *direccionamiento* para esa rama del árbol de temas. Además, los gestores de colas sin suscripciones o publicadores no necesitan conectarse a los hosts de tema. Esta configuración puede reducir significativamente el número de conexiones entre los gestores de colas del clúster y la cantidad de información que se pasa entre los gestores de colas.

Un clúster de publicación/suscripción de direccionamiento de host de tema se comporta de este modo:

- Los temas se definen manualmente en gestores de cola de *host de tema* individuales del clúster.
- Cuando se realiza una suscripción en un gestor de colas de clúster, se crean suscripciones de proxy sólo en los hosts de tema.
- Cuando una aplicación publica información para un tema, el gestor de colas receptor reenvía la publicación a un gestor de colas que aloja el tema. A continuación, el host de tema envía la publicación a todos los gestores de colas del clúster que tienen suscripciones válidas al tema.

Para obtener una descripción más detallada sobre el direccionamiento de host de tema, consulte Direccionamiento de host de tema en clústeres.

En muchas configuraciones, el direccionamiento de host de tema es una topología más adecuada que el *direccionamiento directo*, ya que ofrece las siguientes ventajas:

- Escalabilidad mejorada de clústeres más grandes. Sólo los gestores de colas de host de tema deben poder conectarse a todos los demás gestores de colas del clúster. Por lo tanto, hay menos canales entre gestores de colas y hay menos tráfico administrativo de publicación/suscripción entre gestores de colas que para el direccionamiento directo. Cuando las suscripciones cambian en un gestor de colas, sólo es necesario informar a los gestores de colas de host de temas.
- Más control sobre la configuración física. Con el direccionamiento directo, todos los gestores de colas asumen todos los roles y, por lo tanto, todos necesitan tener la misma capacidad. Con el direccionamiento de host de tema, puede elegir de forma explícita los gestores de colas de host de tema. Por lo tanto, puede garantizar que esos gestores de colas se ejecutan en equipos adecuados y puede utilizar sistemas menos potentes para los demás gestores de colas.

Sin embargo, el direccionamiento de host de tema también impone determinadas restricciones en el sistema:

- La configuración y el mantenimiento del sistema requieren más planificación que para el direccionamiento directo. Tiene que decidir qué apunta al clúster en el árbol de temas y la ubicación de las definiciones de tema en el clúster.
- Al igual que para los temas direccionados directamente, cuando se define un tema direccionado de host de tema nuevo, la información se envía a los gestores de colas de depósito completo y desde allí directamente a todos los miembros del clúster. Este suceso hace que se inicien canales en cada miembro del clúster desde los depósitos completos si aún no se han iniciado.

- Las publicaciones siempre se envían a un gestor de colas de host desde un gestor de colas no de host, incluso si no existen suscripciones en el clúster. Por lo tanto, debe utilizar los temas direccionados cuando normalmente se espera que existan suscripciones o cuando la sobrecarga de conectividad y conocimiento globales es mayor que el riesgo de tráfico de publicación adicional.
- Los mensajes que se publican en los gestores de colas no de host no van directamente al gestor de colas que aloja la suscripción, siempre se direccionan a través de un gestor de colas de host de tema. Este enfoque puede aumentar la actividad general total en el clúster, aumentar la latencia de mensajes y disminuir el rendimiento.

Nota: En determinadas configuraciones, puede eliminar esta restricción, como se describe en la sección [Direccionamiento de host de tema mediante publicadores o suscriptores centralizados](#).

- La utilización de un gestor de colas de host de tema único presenta un punto único de anomalía para todos los mensajes que se publican en un tema. Puede eliminar este único punto de anomalía definiendo varios hosts de temas. Sin embargo, tener varios hosts afecta el orden de los mensajes publicados recibidos por las suscripciones.
- Los gestores de colas de host de tema causan una carga de mensajes adicional, porque es necesario que procesen el tráfico de publicación de varios gestores de colas. Esta carga se puede reducir: Utilice varios hosts de temas para un solo tema (en cuyo caso no se mantiene el orden de mensajes) o utilice gestores de colas diferentes para alojar los temas direccionados para diferentes ramas del árbol de temas.

Direccionamiento de host de tema con publicadores o suscriptores centralizados

Para eliminar el "salto" adicional que ocurre cuando las publicaciones se direccionan siempre a las suscripciones a través de un gestor de colas de host de tema, configure los publicadores o las suscripciones en el mismo gestor de colas que aloja el tema. Este método ofrece ventajas de máximo rendimiento en los dos casos siguientes:

- Temas con muchos publicadores y pocas suscripciones. En este caso, aloje las suscripciones en el gestor de colas de host de tema.
- Temas con pocos publicadores y muchas suscripciones. En este caso, aloje los publicadores en el gestor de colas de host de tema.

La figura siguiente muestra un gestor de colas de host de tema que también aloja las suscripciones. Este método elimina el "salto" adicional entre el publicador y el suscriptor, y reduce la compartición innecesaria de información de suscripciones entre todos los miembros del clúster.

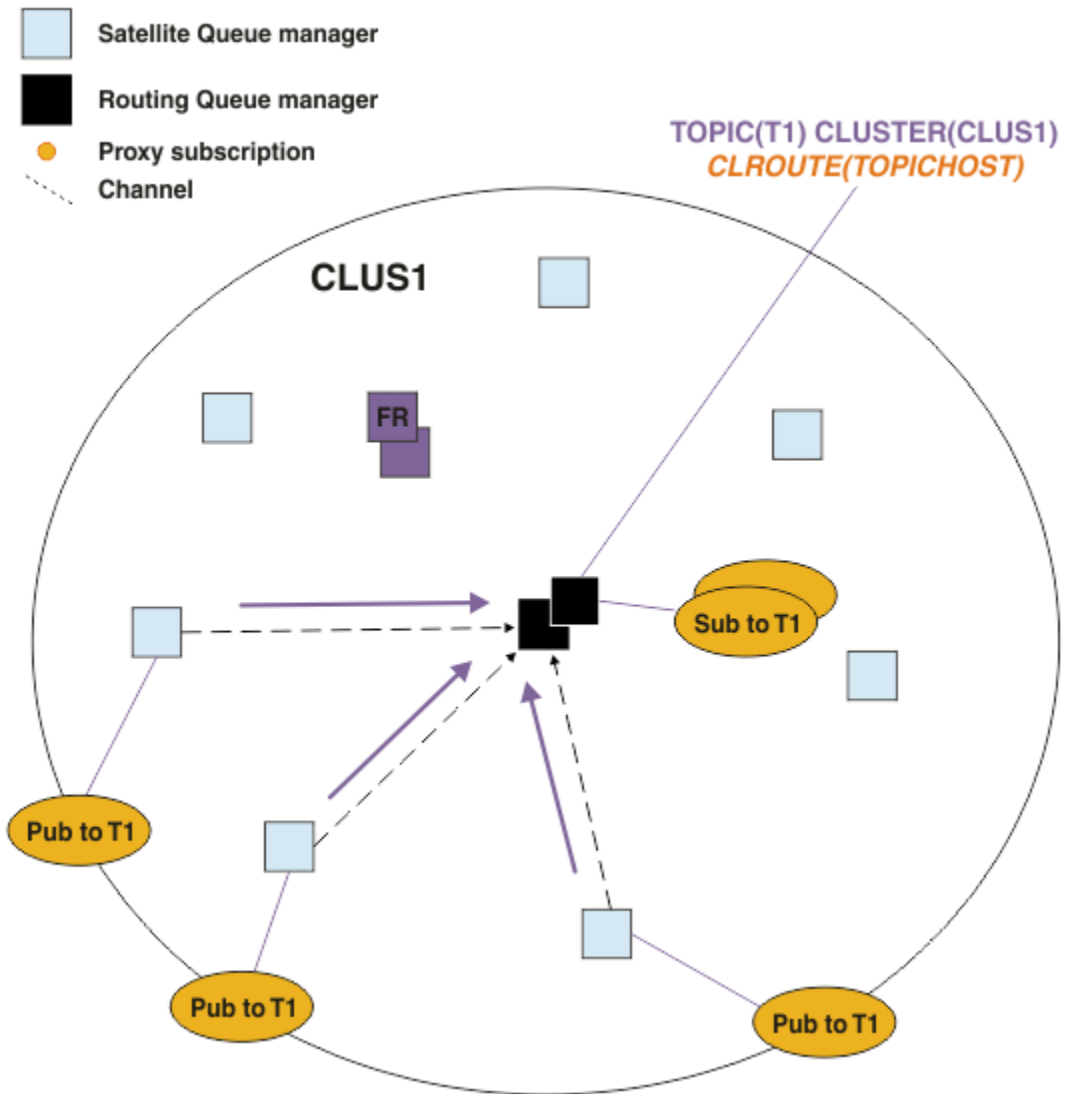


Figura 27. Alojamiento de suscripciones en un gestor de colas de host de tema

La figura siguiente muestra un gestor de colas de host de tema que también aloja a los publicadores. Este método elimina el "salto" adicional entre el publicador y el suscriptor, y reduce la compartición innecesaria de información de suscripciones entre todos los miembros del clúster.

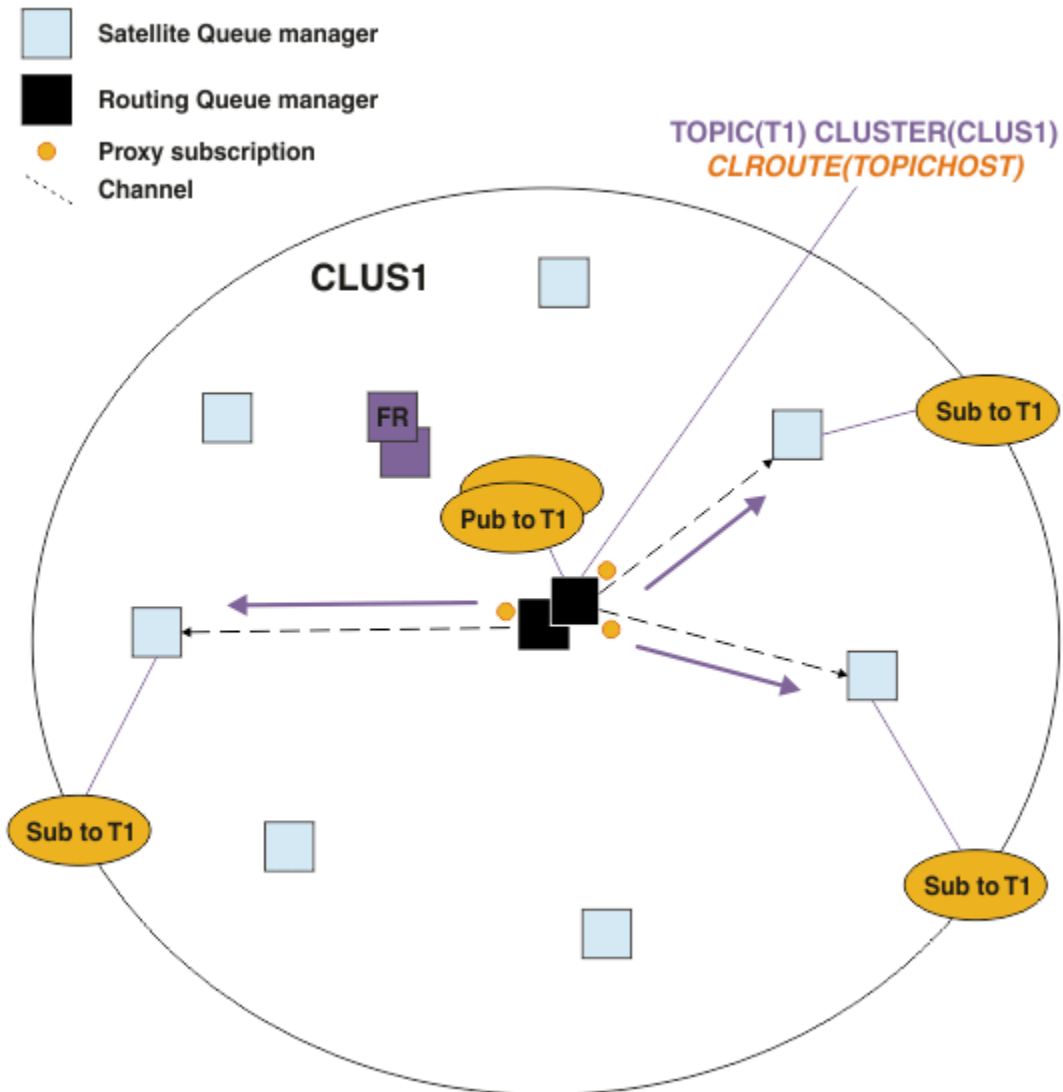


Figura 28. Alojamiento de publicaciones en un gestor de colas de host de tema

Conceptos relacionados

Rendimiento de los clústeres de publicación/suscripción de direccionamiento directo

En los clústeres de publicación/suscripción de direccionamiento directo, la información, tal como los temas en clúster y las suscripciones de proxy, se transfiere a todos los miembros del clúster, independientemente de si todos los gestores de colas del clúster participan de forma activa en la mensajería de publicación/suscripción. Este proceso puede crear una importante carga adicional sobre el sistema. Para disminuir el efecto que la gestión de clústeres puede tener en el rendimiento, realice las actualizaciones en los periodos de tiempo de poca actividad, defina un subconjunto de gestores de colas mucho más pequeño para la publicación/suscripción y convierta dicho subconjunto en un clúster "solapante" o pase a utilizar el direccionamiento de host de tema.

Equilibrio entre productos y consumidores en las redes de publicación/suscripción

Un concepto importante en el rendimiento de la mensajería asíncrona es el *equilibrio*. A menos que los consumidores de mensajes estén equilibrados con los productores de mensajes, existe el riesgo de que se acumule el retraso de los mensajes no consumidos y afecte gravemente al rendimiento de varias aplicaciones.

Rendimiento de las suscripciones en las redes de publicación/suscripción

La publicación/suscripción en IBM MQ funciona de modo que se propaga el conocimiento acerca del lugar donde se han creado las suscripciones a diferentes series de temas en la red del gestor de colas. Esto

permite que el gestor de colas en el que se publica un mensaje identifique los otros gestores de colas que requieren una copia del mensaje publicado, para que coincida con sus suscripciones.

Equilibrio entre productos y consumidores en las redes de publicación/suscripción

Un concepto importante en el rendimiento de la mensajería asíncrona es el *equilibrio*. A menos que los consumidores de mensajes estén equilibrados con los productores de mensajes, existe el riesgo de que se acumule el retraso de los mensajes no consumidos y afecte gravemente al rendimiento de varias aplicaciones.

En una topología de mensajería punto a punto, la relación entre los consumidores de mensajes y los productores de mensajes es fácil de entender. Puede obtener estimaciones de producción y consumo de mensajes, cola por cola y canal por canal. Si hay una falta de equilibrio, los cuellos de botella se identifican fácilmente y luego se subsanan.

Es más difícil de averiguar si los publicadores y los suscriptores están equilibrados en una topología de publicación/suscripción. Comience desde cada suscripción y retroceda hasta los gestores de colas que tienen publicadores sobre el tema. Calcule el número de publicaciones que fluyen a cada suscriptor de cada gestor de colas.

Cada publicación que coincide con una suscripción de un gestor de colas remoto del clúster (según las suscripciones de proxy) se coloca en una cola de transmisión. Si varios gestores de colas remotos tienen suscripciones de proxy para dicha publicación, se colocan varias copias del mensaje en una cola de transmisión y cada una de ellas se destina a diferentes canales de emisor.

En un clúster de publicación/suscripción, estas publicaciones están destinadas a la cola SYSTEM . INTER . QMGR . PUBS en el gestor de colas remoto que aloja las suscripciones. En una jerarquía, cada publicación se destina a la cola SYSTEM . BROKER . DEFAULT . STREAM o a cualquier otra cola de la corriente que figure en SYSTEM . QPUBSUB . QUEUE . NAMELIST de los gestores de colas remotos. Cada gestor de colas procesa los mensajes que llegan a dicha cola y los entrega a las suscripciones correctas de dicho gestor de colas.

Por este motivo, supervise la carga en los siguientes puntos donde pueden producirse cuellos de botella:

- Supervise la carga en las colas de suscripciones individuales.
 - Este cuello de botella implica que la aplicación de suscripción no está consumiendo las publicaciones con la misma rapidez con la que se están publicando.
- Supervise la carga en la cola SYSTEM . INTER . QMGR . PUBS o en las colas de corrientes.
 - Este cuello de botella implica que el gestor de colas está recibiendo publicaciones desde uno o varios gestores de colas remotos con más rapidez de la que puede distribuirlas a las suscripciones locales.
 - Cuando se observa en un gestor de colas de host de tema que utiliza el direccionamiento de host de tema en un clúster, puede considerar la posibilidad de convertir gestores de colas adicionales en hosts de temas, permitiendo de este modo que la carga de trabajo de las publicaciones se equilibre entre los mismos. Sin embargo, esto puede afectar el orden de los mensajes entre publicaciones. Consulte la sección [Direccionamiento de host de tema mediante varios hosts de temas para un solo tema](#).
- Supervise la carga de los canales entre el gestor de colas de publicación y los gestores de colas de suscripción, los cuales se alimentan mediante las colas de transmisión del gestor de colas de publicación.
 - Este cuello de botella implica que uno o varios canales no se están ejecutando o que se están publicando mensajes en el gestor de colas local con más rapidez que la que los canales pueden entregar los mensajes al gestor de colas remoto.
 - Cuando utilice un clúster de publicación/suscripción, puede considerar la posibilidad de definir canales de receptor de clúster adicionales en el gestor de colas de destino. Esto permite equilibrar la carga de trabajo de publicaciones entre los mismos. Sin embargo, esto puede afectar el orden de los mensajes entre publicaciones. Asimismo, considere la posibilidad de pasar a una configuración

de varias colas de transmisión del clúster, ya que de este modo se puede mejorar el rendimiento en determinadas circunstancias.

- Si la aplicación de publicación está utilizando una interfaz de publicación/suscripción en cola, supervise la carga en (a) la cola SYSTEM . BROKER . DEFAULT . STREAM y en cualquier otra cola de corrientes que figure listada en SYSTEM . QPUBSUB . QUEUE . NAMELIST; y (b) la cola SYSTEM . BROKER . DEFAULT . SUBPOINT y cualquier otra cola de subpunto que figure listada en SYSTEM . QPUBSUB . SUBPOINT . NAMELIST.
 - Este cuello de botella implica que las aplicaciones de publicación locales están colocando los mensajes con más rapidez de la que el gestor de colas local puede procesarlos.

Conceptos relacionados

Rendimiento de los clústeres de publicación/suscripción de direccionamiento directo

En los clústeres de publicación/suscripción de direccionamiento directo, la información, tal como los temas en clúster y las suscripciones de proxy, se transfiere a todos los miembros del clúster, independientemente de si todos los gestores de colas del clúster participan de forma activa en la mensajería de publicación/suscripción. Este proceso puede crear una importante carga adicional sobre el sistema. Para disminuir el efecto que la gestión de clústeres puede tener en el rendimiento, realice las actualizaciones en los periodos de tiempo de poca actividad, defina un subconjunto de gestores de colas mucho más pequeño para la publicación/suscripción y convierta dicho subconjunto en un clúster "solapante" o pase a utilizar el direccionamiento de host de tema.

Rendimiento de los clústeres de publicación/suscripción de direccionamiento de host de tema

Un clúster de publicación/suscripción de direccionamiento de host de tema le permite controlar exactamente qué gestores de colas alojan cada tema. Estos hosts de tema se convierten en los gestores de colas de *direccionamiento* para esa rama del árbol de temas. Además, los gestores de colas sin suscripciones o publicadores no necesitan conectarse a los hosts de tema. Esta configuración puede reducir significativamente el número de conexiones entre los gestores de colas del clúster y la cantidad de información que se pasa entre los gestores de colas.

Rendimiento de las suscripciones en las redes de publicación/suscripción

La publicación/suscripción en IBM MQ funciona de modo que se propaga el conocimiento acerca del lugar donde se han creado las suscripciones a diferentes series de temas en la red del gestor de colas. Esto permite que el gestor de colas en el que se publica un mensaje identifique los otros gestores de colas que requieren una copia del mensaje publicado, para que coincida con sus suscripciones.

“Supervisión de clústeres” en la página 318

Dentro de un clúster puede supervisar mensajes de aplicaciones, mensajes de control y registros. Existen consideraciones de supervisión especiales cuando la carga del clúster se equilibra entre dos o más instancias de una cola.

Rendimiento de las suscripciones en las redes de publicación/suscripción

La publicación/suscripción en IBM MQ funciona de modo que se propaga el conocimiento acerca del lugar donde se han creado las suscripciones a diferentes series de temas en la red del gestor de colas. Esto permite que el gestor de colas en el que se publica un mensaje identifique los otros gestores de colas que requieren una copia del mensaje publicado, para que coincida con sus suscripciones.

Este método minimiza el envío de los mensajes publicados a los gestores de colas en los que no existe ninguna suscripción coincidente. Sin embargo, la propagación del conocimiento de las suscripciones puede convertirse en una sobrecarga importante cuando el número de series de temas suscritas es elevado y cambia constantemente debido a que se crean y suprime suscripciones con frecuencia.

Puede influenciar en el rendimiento si ajusta el modo en que fluyen las publicaciones y suscripciones por su red de publicación/suscripción. Si su tráfico de red tiene pocas publicaciones y se crean, suprimen o cambian suscripciones con rapidez, puede detener el flujo de la información de suscripción a todos los gestores de colas y, en su lugar, reenviar todas las publicaciones a todos los gestores de colas de la red. También puede restringir el flujo de las suscripciones y publicaciones del proxy para un tema concreto entre los gestores de colas conectados, puede restringir el flujo de las suscripciones de proxy que contienen comodines y también puede disminuir el número y la naturaleza temporal de las series de temas.

Propagación de las suscripciones individuales y *publicación en todas partes*

La *publicación en todas partes* es una alternativa a la propagación individual de suscripciones. Con la propagación individual, únicamente las publicaciones que tienen una suscripción coincidente en un gestor de colas se reenvían a dicho gestor de colas. Con la *publicación en todas partes*, se reenvían todas las publicaciones a todos los gestores de colas de la red. A continuación, los gestores de colas de recepción entregan las publicaciones que coinciden con las suscripciones locales.

Propagación individual de suscripciones

Este mecanismo genera la menor cantidad de tráfico de publicaciones entre gestores de colas, ya que sólo se envían las publicaciones que coinciden con las suscripciones en un gestor de colas.

No obstante:

- Para cada serie de tema individual al que está suscrito, se envía una suscripción de proxy a los otros gestores de colas de la topología de publicación/suscripción. El conjunto de gestores de colas depende del modelo de direccionamiento que se está utilizando, como se describe en [Planificación de la red de publicación/suscripción distribuida](#).
 - Esta sobrecarga de mensajería puede ser importante si se crean o suprimen miles de suscripciones (por ejemplo, se vuelven a crear todas las suscripciones no duraderas después del reinicio de un gestor de colas) o si el conjunto de suscripciones cambia rápidamente y cada una de ellas es para una serie de temas diferente.
 - El número de gestores de colas a los que se propaga la suscripción de proxy también tiene un efecto el nivel de sobrecarga.
- Las suscripciones de proxy fluyen a otros gestores de colas mediante la mensajería asíncrona. Esto tiene el siguiente efecto:
 - Existe un retardo entre la creación de una suscripción y la creación, entrega y proceso de la suscripción del proxy por parte de los otros gestores de colas.
 - Los mensajes que se publican en estos gestores de colas en dicho intervalo no se entregan a la suscripción remota.

Publicación en todas partes

Con este mecanismo no existe en el sistema ninguna carga adicional por suscripción de proxy a series de temas. Esto significa que la creación, supresión o cambio de las suscripciones no genera una mayor carga de procesos y red.

Tampoco existe ningún retardo entre la creación de una suscripción y el flujo de las publicaciones a un gestor de colas, ya que todas las publicaciones se transmiten a todos los gestores de colas. Por lo tanto, no existe ninguna ventana en la que no se entreguen publicaciones a las suscripciones remotas creadas recientemente.

No obstante:

- Si todas las publicaciones se envían a todos los gestores de colas de la topología de publicación/suscripción, se puede generar un tráfico de red excesivo en el que las publicaciones no tengan suscripciones coincidentes en cada gestor de colas.
 - Cuanto mayor sea el número de gestores de colas de la topología, mayor será la sobrecarga.

Debe considerar la posibilidad de utilizar el mecanismo de *publicación en todas partes* cuando espera que la suscripción a una publicación proceda de un número importante de gestores de colas del clúster o de la jerarquía, o en los casos en los que las sobrecargas debidas a suscripciones de proxy sean demasiado importantes ya que se realizan cambios de suscripciones con frecuencia. Debe utilizar el reenvío de suscripciones de proxy individuales en los casos en los que el tráfico de mensajería aumente cuando las publicaciones se envían a todos los gestores de colas, en lugar de enviarlas a los gestores de colas que coinciden con las suscripciones.

Puede establecer el comportamiento de *publicación en todas partes* en cualquier nivel del árbol de temas. Para habilitar *publicar en todas partes*, establezca el parámetro **PROXYSUB** en FORCE para un objeto de tema de alto nivel. Esto genera una suscripción de proxy de comodín única que coincide con todos los temas debajo de este objeto de tema en el árbol de temas. Cuando se establece en un objeto

de tema en clúster, el atributo **PROXYSUB (FORCE)** se propaga a cada gestor de colas de la red, no sólo al gestor de colas en el que se ha definido el tema.

Nota: Cuando se utiliza en una jerarquía, se establece **PROXYSUB (FORCE)** individualmente en cada gestor de colas, por lo que el mecanismo de topología limita naturalmente el número de canales. Sin embargo, cuando se utiliza en un clúster, es posible que se inicien muchos canales:

- En un clúster de direccionamiento de host de tema, los canales se inician desde cada gestor de colas a cada gestor de colas de host de tema.
- En un clúster de direccionamiento directo, los canales se inician desde cada gestor de colas a cada uno de los otros gestores de colas.

La sobrecarga que implica iniciar muchos canales se resalta más en un clúster de direccionamiento directo y puede ocasionar problemas de rendimiento. Consulte [“Rendimiento de los clústeres de publicación/suscripción de direccionamiento directo”](#) en la página 382.

Otros modos de restringir el flujo de suscripciones de proxy y publicaciones entre gestores de colas conectados

Consolidar series de temas

Si se utilizan muchas series de temas diferentes y temporales se crea un nivel de sobrecarga de gestión en cada gestor de colas del sistema al que están adjuntos los publicadores o las suscripciones. Periódicamente debe evaluar el uso de las series de temas para ver si se pueden consolidar. Disminuir la naturaleza temporal y el número de series de temas y, por lo tanto, el número de publicadores y suscripciones para los mismos, disminuye el impacto en el sistema.

Restringir el ámbito de publicación y suscripción

Para un tema determinado, puede utilizar los valores [Ámbito de publicación](#) y [Ámbito de suscripción](#) para mantener las publicaciones y suscripciones locales en el gestor de colas en el que están definidas.

Bloquear las suscripciones a temas con comodines

Puede restringir el flujo de suscripciones de proxy que contienen caracteres comodín estableciendo el atributo de **Topic WILDCARD** en **BLOCK**. Consulte [Comodines en suscripciones de proxy](#).

Consulte también [“Equilibrio entre productos y consumidores en las redes de publicación/suscripción”](#) en la página 388

Supervisión del tráfico de suscripciones de proxy en los clústeres

Cuando considere la carga del sistema debida al tráfico de suscripciones de proxy, además de supervisar las colas que figuran en la sección [“Equilibrio entre productos y consumidores en las redes de publicación/suscripción”](#) en la página 388, supervise las siguientes colas del clúster:

- La cola SYSTEM.INTER.QMGR.FANREQ en el gestor de colas del suscriptor.
- La cola SYSTEM.INTER.QMGR.CONTROL en todos los demás gestores de colas del clúster.

Cualquier acumulación de mensajes importantes que ocurra en estas colas implica que el índice de cambios de suscripciones es demasiado elevado para el sistema o que un gestor de colas no está funcionando correctamente en el clúster. Si sospecha que el problema reside en un gestor de colas específico, compruebe que el soporte de publicación/suscripción no esté inhabilitado para dicho gestor de colas. Consulte **PSMODE** en [ALTER QMGR](#).

Conceptos relacionados

[Rendimiento de los clústeres de publicación/suscripción de direccionamiento directo](#)

En los clústeres de publicación/suscripción de direccionamiento directo, la información, tal como los temas en clúster y las suscripciones de proxy, se transfiere a todos los miembros del clúster, independientemente de si todos los gestores de colas del clúster participan de forma activa en la mensajería de publicación/suscripción. Este proceso puede crear una importante carga adicional sobre el sistema. Para disminuir el efecto que la gestión de clústeres puede tener en el rendimiento, realice las actualizaciones en los periodos de tiempo de poca actividad, defina un subconjunto de gestores de

colas mucho más pequeño para la publicación/suscripción y convierta dicho subconjunto en un clúster "solapante" o pase a utilizar el direccionamiento de host de tema.

Rendimiento de los clústeres de publicación/suscripción de direccionamiento de host de tema

Un clúster de publicación/suscripción de direccionamiento de host de tema le permite controlar exactamente qué gestores de colas alojan cada tema. Estos hosts de tema se convierten en los gestores de colas de *direccionamiento* para esa rama del árbol de temas. Además, los gestores de colas sin suscripciones o publicadores no necesitan conectarse a los hosts de tema. Esta configuración puede reducir significativamente el número de conexiones entre los gestores de colas del clúster y la cantidad de información que se pasa entre los gestores de colas.

Equilibrio entre productos y consumidores en las redes de publicación/suscripción

Un concepto importante en el rendimiento de la mensajería asíncrona es el *equilibrio*. A menos que los consumidores de mensajes estén equilibrados con los productores de mensajes, existe el riesgo de que se acumule el retraso de los mensajes no consumidos y afecte gravemente al rendimiento de varias aplicaciones.

Suscripciones de proxy en una red de publicación/suscripción

Reducción del número de temas no deseados en el árbol de temas

El rendimiento de un sistema de publicación/suscripción se mejora reduciendo el número de temas no deseados en el árbol de temas. ¿Qué es un tema no deseado y cómo se elimina?

Puede crear un gran número de temas sin que ello afecte negativamente al rendimiento. No obstante, algunas maneras de utilizar la publicación/suscripción causan la expansión continua de árboles de temas. Se crea un número excepcionalmente grande de temas y no se vuelven a utilizar más. El número creciente de temas puede convertirse en un problema de rendimiento.

¿Cómo puede evitar diseños que conduzcan a un número elevado y creciente de temas no deseados?
¿Qué puede hacer para ayudar al gestor de colas a eliminar temas no deseados del árbol de temas?

El gestor de colas reconoce un tema no deseado porque no se ha utilizado durante 30 minutos. El gestor de colas elimina los temas no utilizados del árbol de temas automáticamente. La duración de 30 minutos se puede cambiar alterando el atributo del gestor de colas, **TREELIFE**. Puede ayudar al gestor de colas a eliminar temas no deseados asegurándose de que el tema aparezca como no utilizado ante el gestor de colas. La sección "¿Qué es un tema no utilizado?" en la página 392 explica lo que es un tema no utilizado.

Un programador, al diseñar cualquier aplicación, y especialmente al diseñar una aplicación de larga ejecución, considera su uso de recursos: cuántos recursos necesita el programa, si hay demanda ilimitada, y si existen fugas de recursos. Los temas son un recurso que los programas de publicación/suscripción utilizan. Examine la utilización de temas como cualquier otro recurso que un programa utiliza.

¿Qué es un tema no utilizado?

Antes de definir lo que es un tema no utilizado, ¿qué se considera exactamente como un tema?

Cuando una serie de tema, como por ejemplo USA/Alabama/Auburn, se convierte en un tema, el tema se añade al árbol de temas. Se crean nodos de temas adicionales y sus temas correspondientes en el árbol, si es necesario. La serie de tema USA/Alabama/Auburn se convierte en un árbol con tres temas.

- USA
- USA/Alabama
- USA/Alabama/Auburn

Para visualizar todos los temas del árbol de temas, utilice el mandato **runmqsc DISPLAY TPSTATUS(' # ') TYPE(TOPIC)**.

Un tema no utilizado en el árbol de temas tiene las propiedades siguientes.

No está asociado con un objeto de tema

Un objeto de tema administrativo tiene una serie de tema que lo asocia con un tema. Cuando se define el objeto de tema Alabama, si el tema, USA/Alabama, con el que se va a asociar no existe,

el tema se crea a partir de la serie de tema Si el tema existe, el objeto de tema y el tema se asocian conjuntamente utilizando la serie de tema.

No tiene una publicación retenida

Un tema con una publicación retenida resulta de un publicador que coloca un mensaje en un tema con la opción MQPMO_RETAIN.

Utilice el mandato `DISPLAY TPSTATUS('USA/Alabama') RETAINED` de **runmqsc** para comprobar si USA/Alabama tiene una publicación retenida. La respuesta es YES o NO.

Utilice el mandato `runmqsc CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)` para eliminar una publicación retenida de USA/Alabama.

No tiene ningún tema hijo

USA/Alabama/Auburn es un tema sin temas hijo. USA/Alabama/Auburn es el tema hijo directo de USA/Alabama.

Visualice los hijos directos de USA/Alabama con el mandato `DISPLAY TPSTATUS('USA/Alabama/+')` de **runmqsc**.

No hay publicadores activos en el nodo

Un publicador activo en un nodo es una aplicación que tiene el tema abierto para salida.

Por ejemplo, una aplicación abre el objeto de tema denominado **Alabama** con opciones abiertas MQOO_OUTPUT.

Para visualizar los publicadores activos en USA/Alabama y todos sus hijos, utilice el mandato `runmqsc DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN`.

No hay ningún suscriptor activo en el nodo

Un suscriptor activo puede ser una suscripción duradera, o una aplicación que ha registrado una suscripción a un tema con MQSUB, y no la ha cerrado.

Para visualizar las suscripciones activas en USA/Alabama, utilice el mandato `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN` de **runmqsc**.

Para visualizar las suscripciones activas a USA/Alabama y a todos sus hijos, utilice el mandato `runmqsc DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN`.

Gestión del número de temas en un árbol de temas

En resumen, hay varias maneras de gestionar el número de temas en un árbol de temas.

Visualice TPCOUNT

Utilice el mandato `runmqsc DISPLAY PUBSUB ALL` de forma periódica para visualizar la propiedad **TPCOUNT**. Este es el número de nodos de tema del árbol de temas. Si el número aumenta puede indicar que se requiere un TREELIFE más corto o que sea necesario volver a diseñar los temas.

Modifique TREELIFE

De forma predeterminada, un tema no utilizado tiene una duración de 30 minutos. Puede reducir la duración de un tema no deseado.

Por ejemplo, el mandato `runmqsc , ALTER QMGR TREELIFE(900)`, reduce el tiempo de vida de un tema no utilizado de 30 minutos a 15 minutos.

Excepcionalmente, reinicie el gestor de colas

Cuando se reinicia el gestor de colas, el árbol de temas se reinicializa a partir de objetos de tema, nodos con publicaciones retenidas y suscripciones duraderas. Los temas que se han creado por la operación de los programas de publicador y suscriptor se eliminan.

Como último recurso, si el crecimiento en los temas no deseados ha sido la causa de los problemas de rendimiento en el pasado, reinicie el gestor de colas.

Conceptos relacionados

[Árboles de temas](#)

Aspera gateway puede mejorar el rendimiento en redes de alta latencia

El IBM Aspera faspio Gateway proporciona un túnel TCP/IP rápido que puede aumentar significativamente el rendimiento de red para IBM MQ.

Aspera gateway se puede utilizar para mejorar el rendimiento de los canales del gestor de colas. Es especialmente eficaz si la red tiene una latencia alta o tiende a perder paquetes y, normalmente, se utiliza para acelerar la conexión entre gestores de colas en distintos centros de datos.

Sin embargo, para una red rápida que no pierde paquetes hay una disminución en el rendimiento cuando se utiliza Aspera gateway, por lo que es importante comprobar el rendimiento de la red antes y después de definir una conexión de Aspera gateway.

Un gestor de colas que se ejecuta en cualquier plataforma autorizada puede conectarse a través de un Aspera gateway. La propia pasarela se despliega en Red Hat® o Ubuntu Linux, o Windows.

Para obtener más información, consulte [Definición de una conexión de Aspera gateway en Linux o Windows](#).

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos.

Es posible que IBM no ofrezca los productos, servicios o las características que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar podrá utilizarse cualquier producto, programa o servicio equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. El suministro de este documento no le otorga ninguna licencia sobre estas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM de su país o envíe las consultas por escrito a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones contradigan la legislación vigente: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INCUMPLIMIENTO, COMERCIALIZACIÓN O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM puede realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias hechas en esta publicación a sitios web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen de modo alguno un aval de esos sitios web. Los materiales de estos sitios web no forman parte de los materiales para este producto IBM, por lo que la utilización de dichos sitios web es a cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que el usuario le proporcione del modo que considere apropiado sin incurrir por ello en ninguna obligación con respecto al usuario.

Los titulares de licencias de este programa que deseen información del mismo con el fin de permitir: (i) el intercambio de información entre los programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo, en algunos casos, el pago de una cantidad.

El programa bajo licencia que se describe en esta información y todo el material bajo licencia disponible para el mismo lo proporciona IBM bajo los términos del Acuerdo de cliente de IBM, el Acuerdo de licencia de programas internacional de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se han obtenido en un entorno controlado. Por consiguiente, los resultados obtenidos en otros entornos operativos pueden variar de manera significativa. Es posible que algunas mediciones se hayan realizado en sistemas en nivel de desarrollo y no existe ninguna garantía de que estas mediciones serán las mismas en sistemas disponibles generalmente. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se obtuvo de los proveedores de esos productos, sus anuncios publicados u otras fuentes de disponibilidad pública. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad o alguna reclamación relacionada con productos que no sean de IBM. Todas las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones relacionadas con una futura intención o tendencia de IBM están sujetas a cambios o se pueden retirar sin previo aviso y sólo representan metas y objetivos.

Este documento contiene ejemplos de datos e informes que se utilizan diariamente en la actividad de la empresa. Para ilustrar los ejemplos de la forma más completa posible, éstos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es puramente casual.

LICENCIA DE DERECHOS DE AUTOR:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pagar ninguna cuota a IBM para fines de desarrollo, uso, marketing o distribución de programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. IBM, por tanto, no puede garantizar la fiabilidad, servicio o funciones de estos programas.

Puede que si visualiza esta información en copia software, las fotografías e ilustraciones a color no aparezcan.

Información acerca de las interfaces de programación

La información de interfaz de programación, si se proporciona, está pensada para ayudarle a crear software de aplicación para su uso con este programa.

Este manual contiene información sobre las interfaces de programación previstas que permiten al cliente escribir programas para obtener los servicios de IBM MQ.

Sin embargo, esta información puede contener también información de diagnóstico, modificación y ajustes. La información de diagnóstico, modificación y ajustes se proporciona para ayudarle a depurar el software de aplicación.

Importante: No utilice esta información de diagnóstico, modificación y ajuste como interfaz de programación porque está sujeta a cambios.

Marcas registradas

IBM, el logotipo de IBM , ibm.com, son marcas registradas de IBM Corporation, registradas en muchas jurisdicciones de todo el mundo. Hay disponible una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas.

Microsoft y Windows son marcas registradas de Microsoft Corporation en Estados Unidos y/o otros países.

UNIX es una marca registrada de Open Group en Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos y en otros países.

Este producto incluye software desarrollado por Eclipse Project (<https://www.eclipse.org/>).

Java y todas las marcas registradas y logotipos son marcas registradas de Oracle o sus afiliados.



Número Pieza:

(1P) P/N: