

9.4

Monitoraggio e prestazioni per IBM MQ

IBM

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in [“Informazioni particolari” a pagina 397](#).

Questa edizione si applica alla versione 9 release 4 di IBM® MQ e a tutte le successive release e modifiche se non diversamente indicato nelle nuove edizioni.

Quando si inviano informazioni a IBM, si concede a IBM un diritto non esclusivo di utilizzare o distribuire le informazioni in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

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

Indice

Monitoraggio e prestazioni.....	5
Monitoraggio della tua rete IBM MQ.....	5
Integrazione OpenTelemetry.....	5
Funzionamento del controllo di integrità del gestore code.....	6
Monitoraggio eventi.....	9
Monitoraggio di messaggi.....	59
Messaggi di account e statistiche.....	140
Traccia attività applicazione.....	207
Argomenti di sistema per il monitoraggio e la traccia dell'attività.....	296
Monitoraggio in tempo reale.....	307
Monitoraggio dei cluster.....	319
Monitoraggio del bilanciamento dell'applicazione.....	322
Monitoring performance and resource usage on z/OS.....	325
Ottimizzazione della rete IBM MQ.....	381
Ottimizzazione dei canali di connessione client e server.....	381
Ottimizzazione delle reti di pubblicazione / sottoscrizione distribuite.....	383
Riduzione del numero di argomenti indesiderati nella struttura ad albero degli argomenti.....	393
Aspera gateway può migliorare le prestazioni su reti ad alta latenza.....	395
Informazioni particolari.....	397
Informazioni sull'interfaccia di programmazione.....	398
Marchi.....	398

IBM MQ Monitoraggio e prestazioni

Utilizzare le informazioni di controllo e la guida in questa sezione e i suggerimenti di ottimizzazione specifici per migliorare le prestazioni della rete di gestori code.

Informazioni su questa attività

In base alla dimensione e alla complessità della rete del proprio gestore code, è possibile ottenere una serie di informazioni dal monitoraggio della rete. È possibile utilizzare tali informazioni, insieme alle informazioni fornite in specifici suggerimenti di ottimizzazione, per ottimizzare le prestazioni di rete.

Monitoraggio della tua rete IBM MQ

In IBM MQ sono disponibili diverse tecniche di controllo per ottenere statistiche e altre informazioni specifiche su come è in esecuzione la rete del gestore code. Utilizzare le informazioni di controllo e la guida in questa sezione per migliorare le prestazioni della rete del gestore code.

Il seguente elenco fornisce esempi di motivi per il controllo della rete di gestori code:

- Rileva problemi nella rete del gestore code.
- Assistenza nella determinazione delle cause dei problemi nella rete del gestore code.
- Migliora l'efficienza della tua rete di gestori code.
- Familiarizzare con l'esecuzione della rete del gestore code.
- Confermare che la rete del gestore code sia in esecuzione correttamente.
- Generare messaggi quando si verificano determinati eventi.
- Registrare l'attività del messaggio.
- Determinare l'ultima ubicazione nota di un messaggio.
- Controllare varie statistiche di una rete di gestori code in tempo reale.
- Generare un audit trail.
- Account per l'utilizzo delle risorse dell'applicazione.
- Pianificazione della capacità.

▼ 9.4.0

▶ ALW

Integrazione OpenTelemetry

È possibile integrare IBM MQ con un sistema di traccia OpenTelemetry .

Traccia OpenTelemetry

La traccia OpenTelemetry consente di osservare il funzionamento delle applicazioni all'interno di un flusso di dati. Il flusso di dati può, e spesso, incorporare diverse applicazioni. La traccia può mostrare l'intero percorso e fornire informazioni dettagliate sul funzionamento di ogni singola applicazione. IBM MQ fornisce un servizio di traccia che consente di eseguire l'integrazione con un sistema di traccia OpenTelemetry .

Il servizio di traccia IBM MQ OpenTelemetry è implementato come un'uscita API IBM MQ . Viene fornito come programma di supporto IBM , che significa che si dispone della titolarità per utilizzare il servizio di traccia e ricevere supporto come parte della titolarità IBM MQ . Tenere presente che la titolarità IBM MQ consente solo l'utilizzo del componente di uscita IBM MQ Tracing del programma di supporto IBM Instana .

I problemi derivanti dall'utilizzo dell'exit devono essere riportati al supporto IBM MQ a meno che non si stia utilizzando l'exit di traccia con un sistema di monitoraggio IBM Instana e la titolarità IBM Instana .

L'uscita può essere scaricato qui: <https://ibm.biz/mqinstanaexit>.

Installazione e configurazione del servizio di traccia IBM MQ Open Telemetry

I dettagli completi su come installare e configurare la traccia IBM MQ sono disponibili qui: [IBM MQ Tracing](#).

I dettagli delle piattaforme supportate per l'uscita IBM Instana IBM MQ sono disponibili qui: [Supported platforms for on-premises IBM MQ](#).

Multi

Funzionamento del controllo di integrità del gestore code

Il gestore code esegue controlli di integrità periodici per garantire prestazioni stabili e affidabili. Questo argomento descrive alcuni dei controlli di integrità effettuati dal gestore code e spiega come possono essere configurati in base ai requisiti ambientali.

Nella maggior parte degli ambienti, la configurazione predefinita è adatta e non è necessario modificare la frequenza con cui vengono eseguiti questi controlli. Anche utilizzando le impostazioni predefinite, può essere utile comprendere il modo in cui il gestore code si comporta quando viene rilevato un problema e i problemi ambientali che potrebbero causare un errore di controllo. Questo argomento ha lo scopo di spiegare alcuni di questi comportamenti.



Avvertenza: Non apportare alcuna modifica alla frequenza di questi controlli a meno che non venga consigliato dal supporto IBM .

I diversi componenti del gestore code utilizzano vari mezzi per rilevare e risolvere le incongruenze e questo argomento non è inteso a descrivere tutti questi meccanismi. Ad esempio, i processi IBM MQ utilizzano vari meccanismi per garantire che altri processi da cui dipendono siano ancora in esecuzione. I comportamenti descritti sono quei comportamenti che vengono eseguiti periodicamente dal controller di esecuzione per individuare situazioni ambientali o altre situazioni impreviste. Il controller di esecuzione è il principale processo IBM MQ che avvia e gestisce la maggior parte degli altri processi del gestore code. Poiché si tratta di controlli periodici, vengono eseguiti a intervalli specifici che possono essere modificati in una certa misura impostando i parametri di ottimizzazione appropriati.

Alcuni dei controlli descritti vengono eseguiti da un thread di controllo integrità dedicato. Se viene rilevato un problema con il thread di controllo dello stato stesso, viene scritto un messaggio di avviso [AMQ5066](#) nei log degli errori del gestore code.

I comportamenti descritti in questo argomento sono soggetti a modifiche nelle release future, ad esempio se si osserva che un valore predefinito differente è più stabile su una particolare piattaforma o configurazione.

Controlli generali dello stato

Il gestore code esegue una varietà di controlli su un intervallo regolare. Per impostazione predefinita, questi controlli vengono eseguiti ogni 10 secondi (in alcuni casi, il controllo consente due cicli prima di riportare un errore, determinando un intervallo di 20 secondi per tali controlli). Durante le verifiche, il gestore code verifica che i diversi processi eseguiti come parte del gestore code siano ancora in esecuzione. Per un gestore code HA nativo, verifica che il gestore code stia replicando correttamente i dati sulle istanze in standby.

Se un controllo critico non riesce in questo momento (ad esempio, se il processo `amqzmc0` non è più in esecuzione), il gestore code non sarà in grado di continuare l'esecuzione. Tuttavia, la maggior parte dei controlli vengono effettuati per riordinare le risorse di sistema che non sono più necessarie e che potrebbero semplicemente causare la scrittura di un messaggio nei log di errori del gestore code.

Nella maggior parte dei casi, non è necessario modificare la frequenza di questi controlli sanitari generali. La maggior parte degli eventi nel gestore code o nell'ambiente operativo viene rilevata istantaneamente senza la necessità del processo di controllo di integrità generale per rilevarli. Questo processo serve come controllo periodico per qualsiasi cosa non rilevata altrove nel gestore code. Se necessario, la frequenza può essere configurata utilizzando il parametro di ottimizzazione **ECHearBeatLen** . Il valore minimo è 10000 millisecondi (10 secondi). Il valore massimo è 60000 millisecondi (60 secondi). Se impostato sul valore massimo di 60000, potrebbe verificarsi un ritardo di due minuti per alcuni controlli.

Verifica che sia stato effettuato l'avanzamento del log

I controlli del gestore code che scrivono nel log vengono eseguiti ad una velocità ragionevole. Questo non è un controllo che le prestazioni del logger siano ottimali, ma è progettato per individuare condizioni che potrebbero richiedere ulteriore attenzione. Ad esempio, se il disco su cui vengono memorizzati i file di log è particolarmente lento o se il gestore code non sta ricevendo tempo CPU sufficiente in un ambiente inserito in un contenitore per eseguire tutto il suo lavoro.

Se questo controllo non riesce, l'azione eseguita dal gestore code dipende dal tipo di gestore code utilizzato:

- Su un gestore code non HA:
 - Viene scritto un FDC `xecl_W_PERFORMANCE_BOTTLENECK`. Ciò può essere utilizzato come un'indicazione che alcune parti del sistema potrebbero richiedere ulteriore attenzione. Il gestore code rimane in esecuzione. Se gli FDC `xecl_W_PERFORMANCE_BOTTLENECK` vengono visualizzati nella directory degli errori, potrebbe essere necessario lavorare con il proprio team di archiviazione o di piattaforma per capire se le risorse di sistema sottostanti sono sufficienti per l'esecuzione di IBM MQ. Se IBM MQ viene eseguito in contenitori su nodi con commit eccessivo, IBM MQ potrebbe non ricevere un tempo CPU pianificato sufficiente per eseguire tutto il carico di lavoro di messaggistica.
 - Da IBM MQ 9.3.0, un messaggio di avviso `AMQ5068W` viene scritto nei log degli errori del gestore code e non viene scritto alcun FDC `xecl_W_PERFORMANCE_BOTTLENECK`. Se nei log vengono visualizzati i messaggi `AMQ5068W`, potrebbe essere necessario lavorare con il tuo team di archiviazione o di piattaforma per capire se le risorse di sistema sottostanti sono sufficienti per l'esecuzione di IBM MQ. Se IBM MQ viene eseguito in contenitori su nodi con commit eccessivo, IBM MQ potrebbe non ricevere un tempo CPU pianificato sufficiente per eseguire tutto il carico di lavoro di messaggistica. Se vengono scritti in successione cinque messaggi di avvertenza `AMQ5068W`, viene scritto un FDC `xecl_W_PERFORMANCE_BOTTLENECK`.
- Su un gestore code a più istanze:
 - Se il controllo dello stato di avanzamento del log non riesce, l'istanza primaria termina. Se un'istanza in standby è disponibile, verrà avviata e diventerà l'istanza primaria.
 - Da IBM MQ 9.3.0, l'istanza primaria controlla se un'istanza in standby è disponibile prima di terminare. Se un gestore code in standby è disponibile per eseguire il failover sull'istanza primaria, viene terminato. Inoltre, nei log degli errori del gestore code viene scritto un messaggio di avviso `AMQ5068W`.
- Su un gestore code HA nativo, questo controllo si comporta come un gestore code non HA.
- Su un RDQM (gestore code di dati replicati), questo controllo funziona allo stesso modo di un gestore code non HA.

È possibile che un problema con l'avanzamento del log IBM MQ sia causato da un problema di prestazioni nel gestore code stesso.

Per impostazione predefinita, questo controllo viene eseguito ogni 60 secondi, anche se il gestore code attende due cicli di controllo prima di eseguire un'azione. Ciò significa che, con le impostazioni predefinite, devono trascorrere due minuti prima che il gestore code scriva un messaggio di errore (o che abbia avuto esito negativo nel caso di un gestore code HA).

Nella maggior parte dei casi, il funzionamento predefinito è adatto anche quando il file system è lento o al gestore code viene assegnata una piccola quantità di tempo CPU perché altri controlli come il blocco dei file (consultare [“Verifica che i blocchi file siano ancora mantenuti” a pagina 8](#)) e l'operazione del file system di base causeranno il failover di un'istanza primaria prima che venga eseguito questo controllo. Se necessario, è possibile configurare la frequenza di questo controllo utilizzando il parametro di ottimizzazione **LivenessHeartBeatLen**. Il valore massimo su cui può essere configurato è 600 secondi (10 minuti). Il valore minimo di 0 ha l'effetto di disabilitare del tutto il controllo. Per un gestore code non HA, l'unico effetto del controllo è un messaggio di avvertenza supplementare nei log degli errori del gestore code. Per un gestore code a più istanze, è possibile configurare **LivenessHeartBeatLen** in modo che un'istanza primaria del gestore code esegua il failover più rapidamente (riducendo il valore) o lentamente (aumentando il valore). Aumentare il valore per ridurre la frequenza del controllo dell'avanzamento del log può essere utile se l'ambiente riscontra occasionalmente un IO del file system

molto lento, ma in cui si preferisce che l'istanza primaria del gestore code rimanga in esecuzione. Ciò potrebbe essere utile se si dispone di applicazioni che non sono progettate per riconnettersi automaticamente all'istanza in standby e che richiedono un intervento manuale per riavviarle.

Nota: Se **ECHearBeatLen** è stato aumentato, ciò influisce sulla tempistica dei controlli **LivenessHeartBeatLen**. I controlli di avanzamento del log vengono eseguiti quando vengono eseguiti i controlli di integrità generali, in modo che la riduzione della frequenza dei controlli di integrità generali (**ECHearBeatLen**) potrebbe comportare che i controlli di avanzamento del file di log vengano eseguiti fino a 30 secondi dopo il **LivenessHeartBeatLen** configurato.

Prestazioni generali del file system di registrazione

► V 9.4.0

Da IBM MQ 9.4.0, viene emesso un messaggio di avviso [AMQ6729W](#) nel log degli errori del gestore code se le operazioni generali di lettura / scrittura in questa memoria sembrano essere più lunghe del previsto. È possibile utilizzare le variabili di ambiente **AMQ_IODELAY** per ottimizzare la diagnostica e le tempistiche per facilitare la diagnosi dei problemi di prestazioni della memoria o per aumentare la tolleranza per tali ritardi. Per ulteriori informazioni, consultare [AMQ_IODELAY](#), [AMQ_IODELAY_INMS](#) e [AMQ_IODELAY_FFST](#).

Verifica che i blocchi file siano ancora mantenuti

Per un gestore code a più istanze, il controller di esecuzione verifica periodicamente che conservi ancora il blocco esclusivo sul file a più istanze primario. In molti casi, se il blocco viene perso a causa di un problema con il server NFS, l'istanza primaria esegue il failover quasi immediatamente (prima che venga eseguito questo controllo). Vengono effettuati ulteriori controlli periodici di blocco file per garantire che il gestore code primario esegua il failover in caso di un problema insolito del file system.

Per impostazione predefinita, questi controlli di blocco file vengono eseguiti ogni 20 secondi. Se necessario, questo valore può essere modificato impostando il parametro di ottimizzazione **FileLockHeartBeatLen**. Il valore predefinito per il parametro di ottimizzazione è 10 secondi (il gestore code consente due cicli di controllo prima di eseguire un'operazione che determina il comportamento predefinito di controllo ogni 20 secondi). Il valore minimo del parametro di ottimizzazione è 10 secondi, il valore massimo è 600 secondi (10 minuti).

Nota: Se il **ECHearBeatLen** è stato aumentato, ciò influisce sulla tempistica dei controlli **FileLockHeartBeatLen**. I controlli di blocco dei file vengono eseguiti quando vengono eseguiti i controlli di integrità generali, in modo che la riduzione della frequenza dei controlli di integrità generali (**ECHearBeatLen**) potrebbe determinare la creazione di controlli di blocco dei file fino a 30 secondi dopo il **FileLockHeartBeatLen** configurato.

Controlli sull'integrità dell'applicazione dell'utente

Il gestore code controlla periodicamente che tutte le applicazioni associate localmente che non sono più in esecuzione abbiano eseguito una chiamata MQDISC MQI prima di terminare. Questi controlli vengono eseguiti contemporaneamente ai controlli di integrità generali descritti in [“Controlli generali dello stato” a pagina 6](#). L'intervallo predefinito per tali controlli è quindi 10000 millisecondi (10 secondi) e la modifica del valore del parametro di ottimizzazione **ECHearBeatLen** modifica la frequenza con cui vengono effettuati. Questo controllo è principalmente per garantire che tutte le risorse associate a un'applicazione connessa vengano liberate, non causa l'arresto di un gestore code HA o non HA o il failover su un'istanza alternativa.

Le applicazioni client IBM MQ che sono state terminate senza emettere una chiamata MQDISC MQI vengono rilevate separatamente dal processo dell'agent e tutte le risorse associate alla connessione vengono rilasciate.

Concetti correlati

[Configurazioni HA \(High Availability\)](#)

Monitoraggio eventi

Il controllo eventi è il processo di rilevamento delle ricorrenze di *eventi di strumentazione* in una rete di gestore code. Un evento di strumentazione è una combinazione logica di eventi rilevati da un gestore code o da un'istanza del canale. Tale evento fa sì che il gestore code o l'istanza del canale inserisci un messaggio speciale, denominato *messaggio evento*, su una coda eventi.

Gli eventi di strumentazione IBM MQ forniscono informazioni su errori, avvertenze e altre ricorrenze significative in un gestore code. Utilizzare questi eventi per monitorare il funzionamento dei gestori code nella rete di gestori code per raggiungere i seguenti obiettivi:

- Rileva problemi nella rete del gestore code.
- Assistenza nella determinazione delle cause dei problemi nella rete del gestore code.
- Generare un audit trail.
- Reagire alle modifiche di stato del gestore code

Riferimenti correlati

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

[Riferimento messaggio evento](#)

[Formato messaggio evento](#)

Pubblicazione dei tuoi messaggi di evento IBM MQ

Come preparare IBM MQ a pubblicare i messaggi di evento.

Informazioni su questa attività

I messaggi di evento vengono scritti in code denominate in modo speciale denominate SYSTEM.ADMIN.<feature name>.EVENT.

La cosa importante da notare su queste code di eventi è che è il nome che conta. Per impostazione predefinita, su un gestore code, tutte le code eventi sono definite come code locali. Tuttavia, è possibile eliminare queste code e ridefinirle, magari come una coda remota, in modo che tutti gli eventi vengano incanalati in un gestore code di elaborazione eventi dedicato. In alternativa, è possibile utilizzare una coda alias che punta a un oggetto argomento.

In entrambi i casi, qualsiasi tecnica di reindirizzamento richiede che le applicazioni che leggono le code di eventi non abbiano codificato il nome della coda da cui leggere. Pertanto, è necessario essere in grado di configurare la coda da cui le applicazioni stanno leggendo.

I comandi seguenti mostrano come è possibile ridefinire le code di eventi in modo che i messaggi di eventi vengano pubblicati, utilizzando i seguenti presupposti. Si dispone di:

- Non avviato utilizzando gli eventi oppure
- Sono stati rimossi tutti i messaggi dalle code eventi esistenti e sono state eliminate le code locali prima di questa procedura.

Questi passi mostrano solo le code eventi QMGR e CHANNEL che vengono ridefinite, ma questo potrebbe essere esteso per tutti gli eventi.

Nota: La stringa di argomenti è progettata in modo che un'applicazione possa essere sottoscritta a tutti gli eventi utilizzando un carattere jolly o a eventi specifici, come richiesto.

Procedura

Immettere i seguenti comandi:

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

```

Supponendo che l'applicazione di lettura eventi sia in grado di leggere i messaggi di evento da qualsiasi coda, tale applicazione può essere riconfigurata per leggere da una delle code definite in precedenza come richiesto.

La configurazione `PSPROP(NONE)` sui comandi `DEFINE SUB` garantisce che nessuna proprietà del messaggio aggiunta dal motore di pubblicazione / sottoscrizione, ad esempio `MQTopicString`, venga aggiunta al messaggio di evento, garantendo che le applicazioni esistenti possano continuare a funzionare senza modifiche.

Inoltre, le applicazioni possono anche sottoscrivere direttamente utilizzando la chiamata `MQSUB` per ricevere le informazioni, come modo alternativo invece di utilizzare il comando di gestione `DEFINE SUB`.

Ora, più applicazioni sono in grado di utilizzare le informazioni emesse negli eventi dal gestore code.

Eventi di strumentazione

Un evento di strumentazione è una combinazione logica di condizioni che un gestore code o un'istanza del canale rileva e inserisce in una coda eventi un messaggio speciale, denominato *messaggio evento*.

Gli eventi di strumentazione IBM MQ forniscono informazioni su errori, avvertenze e altre ricorrenze significative in un gestore code. È possibile utilizzare questi eventi per monitorare il funzionamento dei gestori code (con altri metodi come Tivoli NetView per z/OS).

[Figura 1 a pagina 11](#) illustra il concetto di eventi di strumentazione.

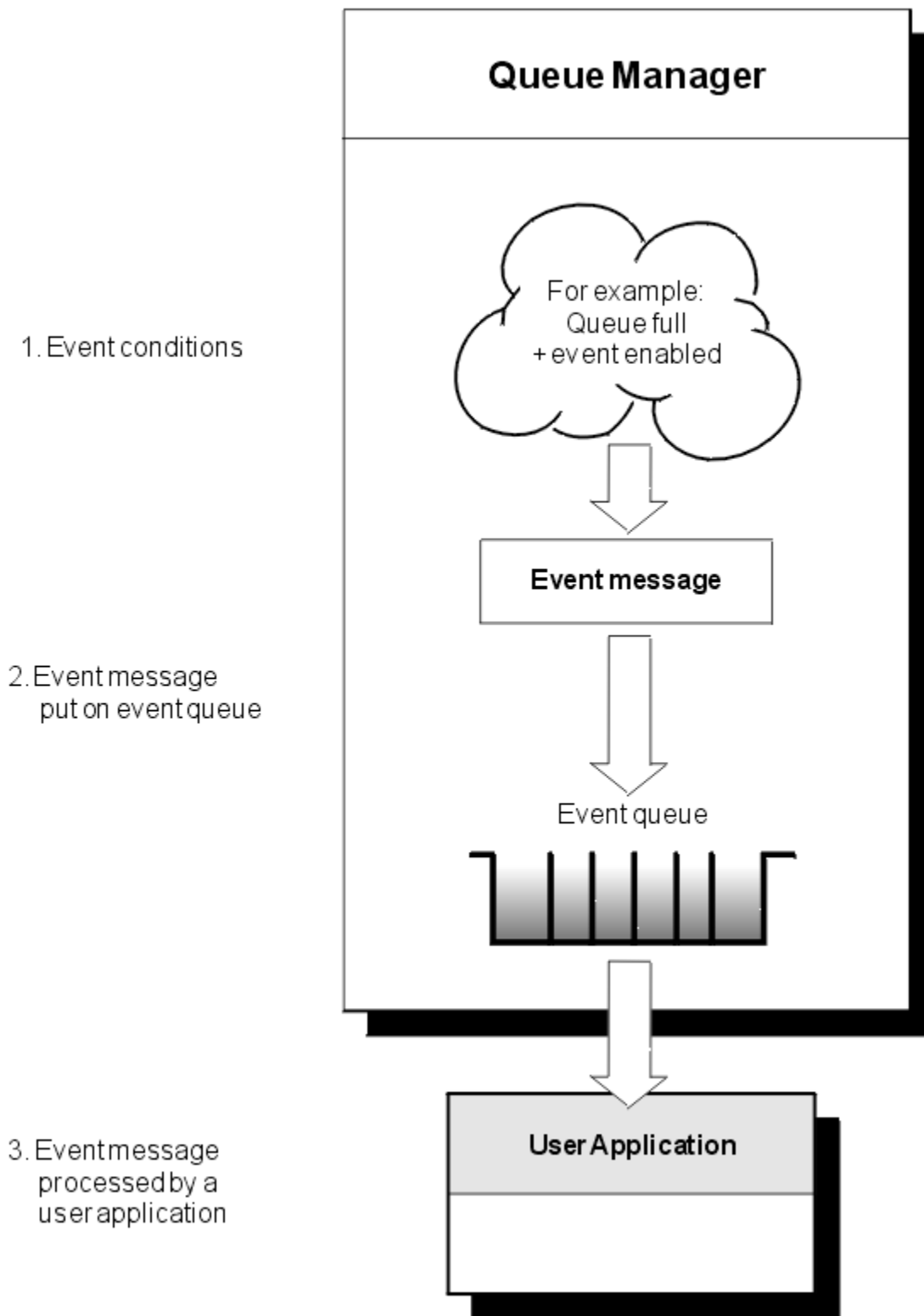


Figura 1. Informazioni sugli eventi di strumentazione

Applicazioni di monitoraggio eventi

Le applicazioni che utilizzano gli eventi per monitorare i gestori code devono includere le seguenti disposizioni:

1. Configurare i canali tra i gestori code nella rete.

2. Implementare le conversioni di dati richieste. Si applicano le normali norme di conversione dei dati. Ad esempio, se si stanno monitorando eventi su un gestore code del sistema UNIX da un gestore code z/OS , assicurarsi di convertire EBCDIC in ASCII.

Notifica eventi tramite code eventi

Quando si verifica un evento, il gestore code inserisce un messaggio di evento nella coda eventi appropriata, se definita. Il messaggio evento contiene informazioni sull'evento che è possibile recuperare scrivendo un programma applicativo MQI adatto che esegue le seguenti operazioni:

- Richiamare il messaggio dalla coda.
- Elaborare il messaggio per estrarre i dati evento.

Le informazioni correlate descrivono il formato dei messaggi evento.

Condizioni che causano eventi

Il seguente elenco fornisce esempi di condizioni che possono causare eventi di strumentazione:

- Viene raggiunto un limite di soglia per il numero di messaggi su una coda.
- Un'istanza del canale è stata avviata o arrestata.
- Un gestore code diventa attivo o viene richiesto l'arresto.
- Un'applicazione tenta di aprire una coda specificando un ID utente non autorizzato sui sistemi IBM MQ for IBM i, AIX, Linux®, and Windows .
- Gli oggetti vengono creati, eliminati, modificati o aggiornati.
- Un comando MQSC o PCF viene eseguito correttamente.
- Un gestore code inizia a scrivere in una nuova estensione di log.
- Inserimento di un messaggio nella coda di messaggi non recapitabili, se le condizioni dell'evento sono soddisfatte.

Concetti correlati

[“Eventi di prestazioni” a pagina 24](#)

Gli eventi delle prestazioni sono correlati alle condizioni che possono influire sulle prestazioni delle applicazioni che utilizzano una specifica coda. L'ambito degli eventi delle prestazioni è la coda. Le chiamate **MQPUT** e **MQGET** su una coda non influiscono sulla generazione di eventi delle prestazioni su un'altra coda.

[“Programma di esempio per monitorare gli eventi di strumentazione su Multiplatforms” a pagina 56](#)
amqsevt formatta gli eventi di strumentazione che un gestore code può creare e viene fornito con IBM MQ for Multiplatforms. Il programma legge i messaggi dalle code eventi e li formatta in stringhe leggibili.

Tipi di evento

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

Gli eventi di strumentazione IBM MQ hanno i tipi seguenti:

- Eventi di gestori code
- Eventi canale e bridge
- Eventi di prestazioni
- Eventi di configurazione
- Eventi di comandi
- Eventi di registrazione
- Eventi locali

Per ogni gestore code, ciascuna categoria di eventi ha la sua coda eventi. Tutti gli eventi in tale categoria risultano in un messaggio di evento inserito nella stessa coda.

Questa coda eventi:

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 messaggi da:

Eventi di gestori code
 Eventi di canale
 Eventi di prestazioni
 Eventi di configurazione
 Eventi di comandi
 Eventi di registrazione
 Richiama gli eventi correlati a Pubblicazione / Sottoscrizione. Utilizzato solo con Multicast.
 Per ulteriori informazioni, vedere [Monitoraggio dell'applicazione multicast](#).

Incorporando gli eventi di strumentazione nella propria applicazione di gestione del sistema, è possibile monitorare le attività in molti gestori code, in molti nodi differenti e per più applicazioni IBM MQ . In particolare, è possibile monitorare tutti i nodi nel proprio sistema da un singolo nodo (per quei nodi che supportano gli eventi IBM MQ) come mostrato in [Figura 2 a pagina 13](#).

Gli eventi di strumentazione possono essere notificati tramite un meccanismo di notifica scritto dall'utente a un'applicazione di amministrazione che può presentare gli eventi a un operatore.

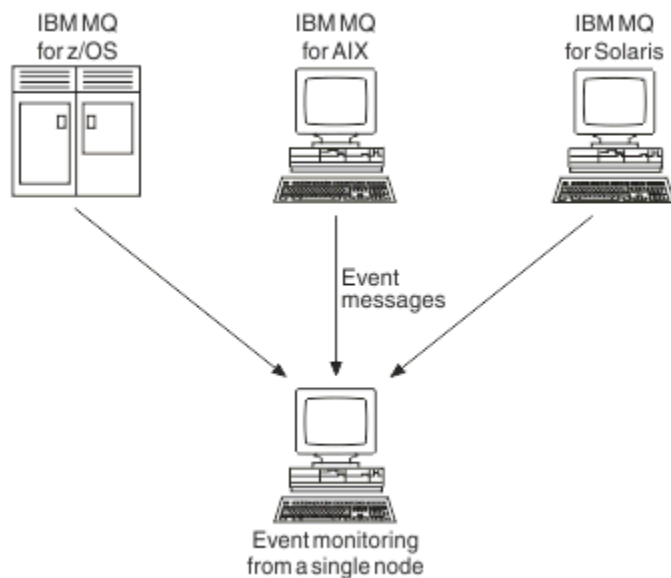


Figura 2. Monitoraggio dei gestori code su diverse piattaforme, su un unico nodo

Gli eventi di strumentazione consentono inoltre alle applicazioni che fungono da agent per altre reti di gestione, ad esempio Tivoli NetView per z/OS, di monitorare i report e creare gli avvisi appropriati.

Eventi di gestori code

Gli eventi del gestore code sono correlati all'utilizzo delle risorse all'interno dei gestori code. Ad esempio, un evento gestore code viene generato se un'applicazione tenta di inserire un messaggio in una coda che non esiste.

I seguenti esempi mostrano le condizioni che possono causare un evento del gestore code:

- Un'applicazione emette una chiamata MQI che non riesce. Il codice di errore dalla chiamata è uguale al codice di errore nel messaggio di evento.

Una condizione simile può verificarsi durante l'operazione interna di un gestore code; ad esempio, quando si genera un messaggio di report. Il codice motivo in un messaggio evento potrebbe

corrispondere a un codice motivo MQI, anche se non è associato ad alcuna applicazione. Non presumere che, poiché un codice di errore del messaggio di evento sembra un codice di errore MQI, l'evento sia stato necessariamente causato da una chiamata MQI non riuscita da un'applicazione.

- Viene immesso un comando per un gestore code e l'elaborazione di questo comando causa un evento. Ad esempio:
 - Un gestore code è stato arrestato o avviato.
 - Viene emesso un comando dove l'ID utente associato non è autorizzato per tale comando.

IBM MQ inserisce i messaggi per gli eventi del gestore code sul sistema SYSTEM.ADMIN.QMGR.EVENT e supporta i seguenti tipi di eventi del gestore code:

ALW Autorizzazione (solo su AIX, Linux, and Windows)

Gli eventi di autorizzazione riportano un'autorizzazione, ad esempio un'applicazione che tenta di aprire una coda per cui non dispone dell'autorizzazione richiesta o un comando emesso da un ID utente che non dispone dell'autorizzazione richiesta. Il messaggio di evento di autorizzazione può contenere i seguenti dati evento:

- [Non autorizzato \(tipo 1\)](#)
- [Non autorizzato \(tipo 2\)](#)
- [Non autorizzato \(Tipo 3\)](#)
- [Non autorizzato \(tipo 4\)](#)
- [Non autorizzato \(tipo 5\)](#)
- [Non autorizzato \(tipo 6\)](#)

Tutti gli eventi di autorizzazioni sono validi solo su AIX, Linux, and Windows .

Inibisci

Gli eventi di inibizione indicano che è stata tentata un'operazione MQPUT o MQGET su una coda in cui la coda non è consentita per le operazioni di inserimento o di ricezione o su un argomento in cui l'argomento non è consentito per le pubblicazioni. Il messaggio di evento di inibizione può contenere i seguenti dati evento:

- [Ricezione inibita](#)
- [Immissione inibita](#)

Locale

Quando un'applicazione o il gestore code non sono stati in grado di accedere a una coda locale o ad un altro oggetto locale, ad esempio, poiché l'oggetto non è stato definito, il gestore code può generare un messaggio di evento locale. Il messaggio evento locale può contenere i seguenti dati evento:

- [Errore tipo coda base alias](#)
- [Coda base alias sconosciuta](#)
- [Nome oggetto sconosciuto](#)

Remoto

Quando un'applicazione o il gestore code non possono accedere a una coda remota su un altro gestore code, ad esempio, la coda di trasmissione non è definita correttamente, il gestore code può generare un messaggio di evento remoto. Il messaggio di evento remoto può contenere i seguenti dati evento:

- [Errore tipo coda di trasmissione predefinita](#)
- [Errore di utilizzo coda di trasmissione predefinita](#)
- [Errore tipo coda](#)
- [Errore nome coda remota](#)
- [Errore tipo coda trasmissione](#)
- [Errore utilizzo coda trasmissione](#)

- [Coda di trasmissione predefinita sconosciuta](#)
- [Gestore code remoto sconosciuto](#)
- [Coda trasmissione sconosciuta](#)

Avvia e arresta

Gli eventi di avvio e arresto indicano che un gestore code è stato avviato o che è stato richiesto di arrestarlo o di disattivarlo.

 z/OS supporta solo eventi di avvio.

Gli eventi di arresto non vengono registrati a meno che non si tratti della persistenza del messaggio predefinito di SYSTEM.ADMIN.QMGR.EVENT è definita come persistente. Il messaggio di evento di avvio e arresto può contenere i seguenti dati evento:

- [Gestore code attivo](#)
- [Gestore code non attivo](#)

Per ciascun tipo di evento in questo elenco, è possibile impostare un attributo del gestore code per abilitare o disabilitare il tipo di evento.

Eventi canale e bridge

I canali riportano questi eventi come risultato di condizioni rilevate durante il loro funzionamento. Ad esempio, quando un'istanza del canale viene arrestata.

Gli eventi del canale vengono generati nelle seguenti circostanze:

- Quando un comando avvia o arresta un canale.
- Quando un'istanza del canale viene avviata o arrestata.
- Quando un canale riceve un'avvertenza di errore di conversione durante la ricezione di un messaggio.
- Quando viene effettuato un tentativo di creare un canale automaticamente, l'evento viene generato se il tentativo ha esito positivo o negativo.

Nota: Le connessioni client non causano eventi Canale avviato o Canale arrestato.

Quando un comando viene utilizzato per avviare un canale, viene generato un evento. Un altro evento viene generato all'avvio dell'istanza del canale. Tuttavia, l'avvio di un canale da parte di un listener, del comando **runmqchl** o di un messaggio di trigger del gestore code non genera un evento. In questi casi, un evento viene generato solo quando l'istanza del canale viene avviata.

Un comando di avvio o arresto del canale con esito positivo genera almeno due eventi. Questi eventi vengono generati per entrambi i gestori code connessi dal canale (purché supportino gli eventi).

Se un evento del canale viene inserito in una coda eventi, una condizione di errore fa sì che il gestore code crei un evento.

I messaggi di evento per gli eventi bridge e canale vengono inseriti nel sistema SYSTEM.ADMIN.CHANNEL.EVENT .

I messaggi di evento del canale possono contenere i seguenti dati evento:

- [Canale attivato](#)
- [Errore di definizione automatica canale](#)
- [Definizione automatica canale OK](#)
- [Errore conversione canale](#)
- [Canale non attivato](#)
- [Canale avviato](#)
- [Canale arrestato](#)
- [Canale arrestato dall'utente](#)
- [Canale bloccato](#)

Eventi bridge IMS (solo z/OS)



Questi eventi vengono riportati quando un bridge IMS viene avviato o arrestato.

I messaggi di evento del bridge IMS possono contenere i seguenti dati evento:

- [Bridge avviato](#)
- [Bridge arrestato](#)

Eventi SSL

L'unico evento TLS è l'evento Errore SSL del canale. Questo evento viene riportato quando un canale che utilizza TLS non riesce a stabilire una connessione TLS.

I messaggi di eventi SSL possono contenere i seguenti dati evento:

- [Errore SSL canale](#)
- [Avvertenza SSL canale](#)

Eventi di prestazioni

Gli eventi delle prestazioni sono notifiche che una risorsa ha raggiunto una condizione di soglia. Ad esempio, è stato raggiunto un limite di profondità della coda.

Gli eventi delle prestazioni sono correlati alle condizioni che possono influire sulle prestazioni delle applicazioni che utilizzano una specifica coda. Non vengono generati per le code eventi stesse.

Il tipo di evento viene restituito nel campo identificativo del comando nei dati del messaggio.

Se un gestore code tenta di inserire un evento del gestore code o un messaggio di evento di prestazioni su una coda eventi e viene rilevato un errore che in genere creerebbe un evento, non viene creato un altro evento e non viene eseguita alcuna azione.

Le chiamate MQGET e MQPUT all'interno di un'unità di lavoro possono generare eventi di prestazioni indipendentemente dal fatto che l'unità di lavoro sia sottoposta a commit o a backout.

I messaggi di evento per gli eventi delle prestazioni vengono inseriti nel SISTEMA SYSTEM.ADMIN.PERFM.EVENT .

Esistono due tipi di evento prestazioni:

Eventi di profondità della coda

Gli eventi di profondità della coda sono relativi al numero di messaggi su una coda; ovvero, quanto è piena o vuota la coda. Questi eventi sono supportati per code condivise. I messaggi di evento di profondità della coda possono contenere i seguenti dati evento:

- [Grandezza coda elevata](#)
- [Grandezza coda bassa](#)
- [Coda piena](#)

Eventi intervallo servizio coda

Gli eventi dell'intervallo di servizio della coda si riferiscono all'elaborazione dei messaggi entro un intervallo di tempo specificato dall'utente. Questi eventi non sono supportati per code condivise.



IBM MQ for z/OS supporta gli eventi di profondità della coda per le code QSGDISP (SHARED), ma non gli eventi di intervallo di servizio. Gli eventi del gestore code e del canale non vengono influenzati dalle code condivise. I messaggi di evento del servizio coda possono contenere i seguenti dati evento:

- [Intervallo servizio coda elevato](#)
- [Intervallo servizio coda - OK](#)

Eventi di configurazione

Gli eventi di configurazione vengono generati quando un evento di configurazione viene richiesto esplicitamente o automaticamente quando un oggetto viene creato, modificato o eliminato.

Un messaggio di evento di configurazione contiene informazioni sugli attributi di un oggetto. Ad esempio, un messaggio di evento di configurazione viene generato se viene creato un oggetto elenco nomi e contiene informazioni sugli attributi dell'oggetto elenco nomi.

I messaggi di evento per gli eventi di configurazione vengono inseriti nel SISTEMA SYSTEM.ADMIN.CONFIG.EVENT .

Gli eventi di configurazione funzionano nel modo seguente:

- Una traccia di controllo delle modifiche alla configurazione del gestore code viene conservata mediante i messaggi di evento, scritti nel SISTEMA SYSTEM.ADMIN.CONFIG.EVENT . È possibile abilitare la scrittura di questi eventi mediante il parametro **CONFIGEV** sul comando ALTER QMGR .
- Questi eventi vengono generati quando un comando DEFINE, ALTER o DELETE agisce su un oggetto o quando viene utilizzata una chiamata MQSET.
- È possibile creare un'immagine di base della configurazione del gestore code corrente utilizzando il comando REFRESH QMGR TYPE (CONFIGEV), che creerà un messaggio di eventi per ciascun oggetto del gestore code. Tenere presente che, poiché questa operazione potrebbe richiedere molto tempo, se si dispone di molti oggetti, è possibile suddividere l'attività in serie di oggetti più piccole utilizzando i qualificatori NAME e OBJECT sul comando.
- Il messaggio dell'evento registra uno dei quattro possibili motivi:
 - MQRC_CONFIG_CHANGE_OBJECT
 - MQRC_CONFIG_CREA_OBJECT
 - MQRC_CONFIG_DELETE_OBJECT
 - MQRC_CONFIG_REFRESH_OBJECT

MQRC_CONFIG_CHANGE_OBJECT, MQRC_CONFIG_CREATE_OBJECT o MQRC_CONFIG_DELETE_OBJECT si verificano per i rispettivi comandi MQSC o PCF che potrebbero essere emessi su un oggetto.

MQRC_CONFIG_REFRESH_OBJECT si verifica per quei messaggi di evento scritti, quando si crea l'immagine di base.

Esistono quattro tipi di evento di configurazione:

Crea eventi oggetto

Gli eventi di creazione oggetto vengono generati quando viene creato un oggetto. Il messaggio dell'evento contiene i seguenti dati evento: Crea oggetto.

Eventi di modifica oggetto

Gli eventi di modifica oggetto vengono generati quando un oggetto viene modificato. Il messaggio evento contiene i seguenti dati evento: Modifica oggetto.

Elimina eventi oggetto

Gli eventi di eliminazione oggetto vengono generati quando un oggetto viene eliminato. Il messaggio evento contiene i seguenti dati evento: Elimina oggetto.

Aggiorna eventi oggetto

Gli eventi dell'oggetto di aggiornamento vengono generati da una richiesta esplicita di aggiornamento. Il messaggio di evento contiene i seguenti dati evento: Aggiorna oggetto.

Eventi di comandi

Gli eventi comando vengono notificati quando un comando MQSC o PCF viene eseguito correttamente.

Un messaggio di evento comando contiene informazioni sull'origine, il contesto e il contenuto di un comando. Ad esempio, un messaggio di evento di comando viene generato con tali informazioni se il comando MQSC, ALTER QLOCAL, viene eseguito correttamente.

I messaggi di evento per gli eventi comando vengono inseriti nel SISTEMA SYSTEM.ADMIN.COMMAND.EVENT .

Gli eventi di comando contengono i seguenti dati evento: Comando.

Multi *Eventi di registrazione*

Gli eventi del programma di registrazione vengono riportati quando un gestore code che utilizza la registrazione lineare inizia a scrivere i record di log in una nuova estensione di log **IBM i**, su IBM i, in un nuovo ricevitore di giornale. **z/OS** Gli eventi del programma di registrazione non sono disponibili con IBM MQ for z/OS

Un messaggio di evento del programma di registrazione contiene informazioni che specificano le estensioni di log richieste dal gestore code per riavviare il gestore code o per il ripristino del supporto.

I messaggi di evento per gli eventi del programma di registrazione vengono inseriti nel SISTEMA SYSTEM.ADMIN.LOGGER.EVENT .

Il messaggio dell'evento del programma di registrazione contiene i seguenti dati evento: Logger.

Riepilogo dati messaggio evento

Utilizzare questo riepilogo per ottenere informazioni sui dati evento che ogni tipo di messaggio evento può contenere.

Tipo di evento	Consultare questi argomenti
Eventi di autorizzazione	Non autorizzato (tipo 1)
	Non autorizzato (tipo 2)
	Non autorizzato (Tipo 3)
	Non autorizzato (tipo 4)
	Non autorizzato (tipo 5)
	Non autorizzato (tipo 6)
Eventi di canale	Canale attivato
	Errore di definizione automatica canale
	Definizione automatica canale OK
	Canale bloccato
	Errore conversione canale
	Canale non attivato
	Canale avviato
	Canale arrestato
Canale arrestato dall'utente	
Eventi di comandi	Comando
Eventi di configurazione	Crea oggetto
	Modifica oggetto
	Elimina oggetto
	Aggiorna oggetto

Tipo di evento	Consultare questi argomenti
Eventi bridge IMS	Bridge avviato
	Bridge arrestato
Eventi di inibizione	Ricezione inibita
	Immissione inibita
Eventi locali	Errore tipo coda base alias
	Coda base alias sconosciuta
	Nome oggetto sconosciuto
Eventi di registrazione	Programma di registrazione
Eventi di prestazioni	Grandezza coda elevata
	Grandezza coda bassa
	Coda piena
	Intervallo servizio coda elevato
	Intervallo servizio coda - OK
Eventi remoti	Errore tipo coda di trasmissione predefinita
	Errore di utilizzo coda di trasmissione predefinita
	Errore tipo coda
	Errore nome coda remota
	Errore tipo coda trasmissione
	Errore utilizzo coda trasmissione
	Coda di trasmissione predefinita sconosciuta
	Gestore code remoto sconosciuto
	Coda trasmissione sconosciuta
Eventi SSL	Errore SSL canale
Eventi di avvio e arresto	Gestore code attivo
	Gestore code non attivo


Controllo degli eventi

Abilitare e disabilitare gli eventi specificando i valori appropriati per il gestore code, gli attributi della coda o entrambi, a seconda del tipo di evento.

È necessario abilitare ogni evento di strumentazione che si desidera generare. Ad esempio, le condizioni che causano un evento Coda piena sono:

- Gli eventi Coda piena sono abilitati per una coda specificata e
- Un'applicazione emette una richiesta MQPUT per inserire un messaggio su tale coda, ma la richiesta non riesce perché la coda è piena.

Abilitare e disabilitare gli eventi utilizzando una delle tecniche riportate di seguito:

- Comandi script IBM MQ (MQSC).
- I comandi PCF IBM MQ corrispondenti.
-  Le operazioni e i pannelli di controllo per gestori code su z/OS.

- IBM MQ Explorer.

Nota: È possibile impostare gli attributi correlati agli eventi sia per code che per gestori code solo tramite comando. La chiamata MQI MQSET non supporta gli attributi correlati agli eventi.

Concetti correlati

[“Eventi di strumentazione” a pagina 10](#)

Un evento di strumentazione è una combinazione logica di condizioni che un gestore code o un'istanza del canale rileva e inserisce in una coda eventi un messaggio speciale, denominato *messaggio evento*.

[Utilizzo delle operazioni e dei pannelli di controllo su z/OS](#)

Attività correlate

[Automazione delle attività di gestione](#)

[Utilizzo dei formati di comando programmabili](#)

Riferimenti correlati

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

[Comandi MQSC](#)

Controllo degli eventi del gestore code

È possibile controllare gli eventi del gestore code utilizzando attributi del gestore code. Per abilitare gli eventi del gestore code, impostare l'attributo appropriato del gestore code su ENABLED. Per disabilitare gli eventi del gestore code, impostare l'attributo del gestore code appropriato su DISABLED.

Per abilitare o disabilitare gli eventi del gestore code, utilizzare il comando MQSC **ALTER QMGR**, specificando l'attributo del gestore code appropriato. Tabella 1 a pagina 20 riepiloga come abilitare gli eventi del gestore code. Per disabilitare un evento del gestore code, impostare il parametro appropriato su DISABLED.

<i>Tabella 1. Abilitazione degli eventi del gestore code utilizzando i comandi MQSC</i>	
Evento	parametro ALTER QMGR
Autorizzazione	AUTHOREV (ABILITATO)
Inibisci	INIBIZIONE (ABILITATA)
Locale	LOCALEV (ABILITATO)
Remoto	REMOOTEV (ABILITATO)
Avvia e arresta	STRSTPEV (ABILITATO)

Controllo degli eventi bridge e canale

Gli eventi del canale vengono controllati utilizzando gli attributi del gestore code. Per abilitare gli eventi del canale, impostare l'attributo del gestore code appropriato su ENABLED. Per disabilitare gli eventi del canale, impostare l'attributo del gestore code appropriato su DISABLED.

Per abilitare o disabilitare gli eventi dei canali utilizzare il comando MQSC **ALTER QMGR**, specificando l'attributo del gestore code appropriato. Tabella 2 a pagina 21 riepiloga come abilitare gli eventi bridge e canale. Per disabilitare un evento del gestore code, impostare il parametro appropriato su DISABLED.


Limitazione:  Gli eventi di definizione automatica del canale non sono disponibili su IBM MQ for z/OS.

Tabella 2. Abilitazione di eventi bridge e canale utilizzando i comandi MQSC

Evento	parametro ALTER QMGR
Canale Correlato solo a errori di canale IMS bridge SSL Definizione automatica canale	CHLEV (ABILITATO) CHLEV (ECCEZIONE) BRIDGEEV (ABILITATO) SSLEV (ABILITATO) CHADEV (ABILITATO)

Con CHLEV impostato su exception, vengono creati i seguenti codici di ritorno e i corrispondenti qualificatori del motivo:

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_ERROR
- MQRC_CHANNEL_NOT_ACTIVATED
- MQRC_CHANNEL_STOPPED
 - con i seguenti ReasonQualifiers:
 - MQRQ_CHANNEL_STOPPED_ERROR
 - MQRQ_CHANNEL_STOPPED_RETRY
 - MQRQ_CHANNEL_STOPPED_DISABLED
- MQRC_CHANNEL_STOPPED_BY_USER
- MQRC_CHANNEL_BLOCKED
 - con i seguenti ReasonQualifiers:
 - MQRQ_CHANNEL_BLOCKED_NOACCESS
 - MQRQ_CHANNEL_BLOCKED_USERID
 - INDIRIZZO_BLOCCO_CANALE_MQRQ

Controllo degli eventi delle prestazioni

Controllare gli eventi delle prestazioni utilizzando l'attributo del gestore code PERFMEV. Per abilitare gli eventi delle prestazioni, impostare PERFMEV su ENABLED. Per disabilitare gli eventi delle prestazioni, impostare l'attributo del gestore code PERFMEV su DISABLED.

Per impostare l'attributo del gestore code PERFMEV su ENABLED, utilizzare il seguente comando MQSC:

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

Per abilitare specifici eventi di prestazioni, impostare l'attributo della coda appropriato. Inoltre, specificare le condizioni che causano l'evento.

Eventi di profondità della coda

Per impostazione predefinita, tutti gli eventi di profondità della coda sono disabilitati. Per configurare una coda per uno qualsiasi degli eventi di profondità della coda:

1. Abilitare gli eventi delle prestazioni sul gestore code.
2. Abilitare l'evento sulla coda richiesta.
3. Impostare i limiti, se necessario, sui livelli appropriati, espressi come percentuale della profondità massima della coda.

Eventi intervallo servizio coda

Per configurare una coda per gli eventi di intervallo di servizio della coda, è necessario:

1. Abilitare gli eventi delle prestazioni sul gestore code.

2. Impostare l'attributo di controllo per un evento Intervallo servizio coda elevato o OK sulla coda come richiesto.
3. Specificare l'intervallo di tempo di servizio impostando l'attributo Q SVCINT per la coda sull'intervallo di tempo appropriato.

Nota: Quando abilitato, è possibile generare un evento di intervallo del servizio coda in qualsiasi momento appropriato, non necessariamente in attesa che venga emessa una chiamata MQI per la coda. Tuttavia, se una chiamata MQI viene utilizzata su una coda per inserire o rimuovere un messaggio, qualsiasi evento delle prestazioni applicabile viene generato in quel momento. L'evento non viene generato quando il tempo trascorso diventa uguale all'intervallo di servizio.

Controllo degli eventi di configurazione, comando e programma di registrazione

Gli eventi di configurazione, comando e logger vengono controllati utilizzando gli attributi del gestore code CONFIGEV, CMDEV e LOGGEREV. Per abilitare questi eventi, impostare l'attributo del gestore code appropriato su ENABLED. Per disabilitare questi eventi, impostare l'attributo del gestore code appropriato su DISABLED.

Eventi di configurazione

Per abilitare gli eventi di configurazione, impostare CONFIGEV su ENABLED. Per disabilitare gli eventi di configurazione, impostare CONFIGEV su DISABLED. Ad esempio, è possibile abilitare gli eventi di configurazione utilizzando il seguente comando MQSC:

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

Eventi di comandi

Per abilitare gli eventi comando, impostare CMDEV su ENABLED. Per abilitare gli eventi di comando per i comandi tranne i comandi DISPLAY MQSC e i comandi Inquire PCF, impostare CMDEV su NODISPLAY. Per disabilitare gli eventi comando, impostare CMDEV su DISABLED. Ad esempio, è possibile abilitare gli eventi comando utilizzando il seguente comando MQSC:

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

Eventi di registrazione

Per abilitare gli eventi del logger, impostare LOGGEREV su ENABLED. Per disabilitare gli eventi del programma di registrazione, impostare LOGGEREV su DISABLED. Ad esempio, è possibile abilitare gli eventi del programma di registrazione utilizzando il seguente comando MQSC:

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

Code eventi

Quando si verifica un evento, il gestore code inserisce un messaggio evento nella coda eventi definita. Il messaggio di evento contiene informazioni sull'evento.

È possibile definire le code eventi, come:

- Code locali
- Code alias
- Definizioni locali delle code remote o come
- Code cluster remote

Se si definiscono tutte le code eventi come definizioni locali della stessa coda remota su un gestore code, è possibile centralizzare le attività di controllo.

È necessario non definire le code di eventi come code di trasmissione, poiché i messaggi di eventi hanno formati incompatibili con il formato del messaggio richiesto per le code di trasmissione.

Le code eventi condivise sono code locali definite con il valore QSGDISP (SHARED).

Per ulteriori informazioni sulla definizione di code condivise su z/OS, consultare [Programmazione delle applicazioni con code condivise](#).

Quando una coda eventi non è disponibile

Se un evento si verifica quando la coda eventi non è disponibile, il messaggio dell'evento viene perso. Ad esempio, se non si definisce una coda eventi per una categoria di eventi, tutti i messaggi di eventi per tale categoria vengono persi. I messaggi di eventi, ad esempio, non vengono salvati nella coda dei messaggi non recapitabili (messaggi non recapitati).

Tuttavia, è possibile definire la coda eventi come coda remota. Quindi, se si verifica un problema sul sistema remoto durante l'inserimento dei messaggi nella coda risolta, il messaggio di evento arriva sulla coda di messaggi non instradabili del sistema remoto.

Una coda eventi potrebbe non essere disponibile per diversi motivi, tra cui:

- La coda non è stata definita.
- La coda è stata eliminata.
- La coda è piena.
- La coda è stata inserita - inibita.

L'assenza di una coda eventi non impedisce il verificarsi dell'evento. Ad esempio, dopo un evento di prestazioni, il Gestore code modifica gli attributi della coda e reimposta le statistiche della coda. Questa modifica si verifica se il messaggio evento viene inserito o meno nella coda eventi delle prestazioni. Lo stesso vale per gli eventi di configurazione e comando.

Utilizzo di code di eventi attivati

È possibile configurare code di eventi con trigger in modo che quando viene generato un evento, il messaggio di evento inserito nella coda eventi avvia un'applicazione di controllo scritta dall'utente. Questa applicazione può elaborare i messaggi di evento e intraprendere l'azione appropriata. Ad esempio, alcuni eventi potrebbero richiedere che un operatore venga informato, altri eventi potrebbero avviare un'applicazione che esegue automaticamente alcune attività di gestione.

Alle code eventi possono essere associate azioni trigger e possono creare messaggi trigger. Tuttavia, se questi messaggi trigger a loro volta causano condizioni che normalmente generano un evento, non viene generato alcun evento. non generando un evento in questa istanza si garantisce che il loop non si verifichi.

Concetti correlati

[“Controllo degli eventi” a pagina 19](#)

Abilitare e disabilitare gli eventi specificando i valori appropriati per il gestore code, gli attributi della coda o entrambi, a seconda del tipo di evento.

[“Formato dei messaggi di eventi” a pagina 23](#)

I messaggi di evento contengono informazioni su un evento e la sua causa. Come altri messaggi IBM MQ , un messaggio di evento ha due parti: un descrittore del messaggio e i dati del messaggio.

[Programmazione di applicazioni con code condivise](#)

[Condizioni per un evento trigger](#)

Riferimenti correlati

[QSGDisp \(MQLONG\)](#)

Formato dei messaggi di eventi

I messaggi di evento contengono informazioni su un evento e la sua causa. Come altri messaggi IBM MQ , un messaggio di evento ha due parti: un descrittore del messaggio e i dati del messaggio.

- Il descrittore del messaggio è basato sulla struttura MQMD.
- I dati del messaggio sono costituiti da un' *intestazione evento* e dai dati evento . L'intestazione dell'evento contiene il codice motivo che identifica il tipo di evento. L'inserimento del messaggio di

evento e di qualsiasi azione successiva non influisce sul codice motivo restituito dalla chiamata MQI che ha causato l'evento. I dati dell'evento forniscono ulteriori informazioni sull'evento.

Di solito, i messaggi di evento vengono elaborati con un'applicazione di gestione del sistema personalizzata per soddisfare i requisiti dell'azienda in cui vengono eseguiti.

Quando i gestori code in un gruppo di condivisione code rilevano condizioni per la generazione di un messaggio di evento, diversi gestori code possono generare un messaggio di evento per la coda condivisa, generando diversi messaggi di evento. Per garantire che un sistema possa correlare più messaggi di evento da gestori code differenti, tali messaggi di evento hanno un identificativo di correlazione univoco (*CorrelId*) impostato nel descrittore del messaggio (MQMD).

Riferimenti correlati

[“MQMD del report di attività \(descrittore del messaggio\)” a pagina 103](#)

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQMD per un report di attività

[“Report attività MQEPH \(intestazione PCF integrata\)” a pagina 107](#)

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQEPH per un report di attività

[“Report attività MQCFH \(intestazione PCF\)” a pagina 108](#)

Utilizzare questa pagina per visualizzare i valori PCF contenuti nella struttura MQCFH per un report di attività

[Riferimento messaggio evento](#)

[Formato messaggio evento](#)

[Messaggio evento MQMD \(descrittore messaggio\)](#)

[Messaggio evento MQCFH \(intestazione PCF\)](#)

[Descrizioni dei messaggi di eventi](#)

Eventi di prestazioni

Gli eventi delle prestazioni sono correlati alle condizioni che possono influire sulle prestazioni delle applicazioni che utilizzano una specifica coda. L'ambito degli eventi delle prestazioni è la coda. Le chiamate **MQPUT** e **MQGET** su una coda non influiscono sulla generazione di eventi delle prestazioni su un'altra coda.

I messaggi di evento delle prestazioni possono essere generati in qualsiasi momento, non necessariamente in attesa che venga emessa una chiamata MQI per la coda. Tuttavia, se si utilizza una chiamata MQI su una coda per inserire o rimuovere un messaggio, in quel momento vengono generati tutti gli eventi delle prestazioni appropriati.

Ogni messaggio di evento delle prestazioni generato viene inserito nella coda, SYSTEM.ADMIN.PERFM.EVENT.

I dati di evento contengono un codice motivo che identifica la causa dell'evento, una serie di statistiche di eventi delle prestazioni e altri dati. I tipi di dati evento che è possibile restituire nei messaggi evento prestazioni sono descritti nel seguente elenco:

- [Grandezza coda elevata](#)
- [Grandezza coda bassa](#)
- Coda piena
- [Intervallo servizio coda elevato](#)
- [Intervallo servizio coda - OK](#)

Gli esempi che illustrano l'utilizzo degli eventi delle prestazioni presuppongono che si impostino gli attributi della coda utilizzando i comandi IBM MQ appropriati (MQSC). In z/OS, è anche possibile impostare gli attributi della coda utilizzando i pannelli di operazioni e controlli per i gestori code.

Riferimenti correlati

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare


Statistiche evento prestazioni

I dati evento delle prestazioni nel messaggio evento contengono le statistiche relative all'evento. Utilizzare le statistiche per analizzare il funzionamento di una coda specificata.

I dati evento nel messaggio evento contengono informazioni sull'evento per i programmi di gestione del sistema. Per tutti gli eventi di prestazioni, i dati evento contengono il nome del Gestore code e la coda associata all'evento. I dati evento contengono anche statistiche relative all'evento. [Tabella 3 a pagina 25](#) riepiloga le statistiche di eventi che è possibile utilizzare per analizzare il comportamento di una coda. Tutte le statistiche si riferiscono a ciò che è accaduto dall'ultima volta che sono state reimpostate.

Parametro	Descrizione
TimeSinceReset	Il tempo trascorso dall'ultima reimpostazione delle statistiche.
HighQDepth	Il numero massimo di messaggi nella coda dall'ultima reimpostazione delle statistiche.
MsgEnqCount	Il numero di messaggi accodati (il numero di chiamate MQPUT alla coda), dall'ultima reimpostazione delle statistiche.
MsgDeqCount	Il numero di messaggi non accodati (il numero di chiamate MQGET alla coda), dall'ultima reimpostazione delle statistiche.

Le statistiche dell'evento prestazioni vengono reimpostate quando si verifica una delle seguenti modifiche:

- Si verifica un evento prestazioni (le statistiche vengono reimpostate su tutti i gestori code attivi).
- Un gestore code viene arrestato e riavviato.
- Il comando PCF, Reset Queue Statistics, viene emesso da un programma applicativo.
-  Solo su z/OS, il comando RESET QSTATS viene emesso nella console.

Concetti correlati

[“Eventi di prestazioni” a pagina 24](#)

Gli eventi delle prestazioni sono correlati alle condizioni che possono influire sulle prestazioni delle applicazioni che utilizzano una specifica coda. L'ambito degli eventi delle prestazioni è la coda. Le chiamate **MQPUT** e **MQGET** su una coda non influiscono sulla generazione di eventi delle prestazioni su un'altra coda.

[“Il timer del servizio” a pagina 27](#)

Gli eventi dell'intervallo di servizio coda utilizzano un timer interno, denominato *timer servizio*, controllato dal gestore code. Il timer del servizio viene utilizzato solo se è abilitato un evento dell'intervallo del servizio della coda.

[“Regole per gli eventi dell'intervallo di servizio della coda” a pagina 27](#)

Le regole formali controllano quando il timer del servizio è impostato e vengono generati gli eventi di intervallo del servizio della coda.

Attività correlate

[“Abilitazione degli eventi di intervallo di servizio della coda” a pagina 28](#)

Per configurare una coda per gli eventi dell'intervallo di servizio della coda, è necessario impostare il gestore code e gli attributi della coda appropriati.

Riferimenti correlati

[Grandezza coda elevata](#)

[Reimposta statistiche coda](#)

[QSTATS RESET](#)

Eventi intervallo servizio coda

Gli eventi dell'intervallo di servizio della coda indicano se un'operazione è stata eseguita su una coda all'interno di un intervallo di tempo definito dall'utente denominato *intervallo di servizio*. A seconda dell'installazione, è possibile utilizzare gli eventi dell'intervallo di servizio della coda per monitorare se i messaggi vengono rimossi dalle code abbastanza rapidamente.

Gli eventi dell'intervallo di servizio della coda non sono supportati sulle code condivise.

I seguenti tipi di eventi dell'intervallo di servizio della coda possono verificarsi, dove il termine *operazione get* fa riferimento a una chiamata **MQGET** o a un'attività che rimuove un messaggio da una coda, ad esempio utilizzando il comando **CLEAR QLOCAL** :

Intervallo servizio coda - OK

Indica che dopo una delle seguenti operazioni:

- Una chiamata MQPUT
- Un'operazione get che lascia una coda non vuota

un'operazione get è stata eseguita entro un periodo di tempo definito dall'utente, noto come *intervallo di servizio*.

Solo un'operazione di richiamo può causare il messaggio di evento Intervallo servizio coda OK. Gli eventi OK dell'intervallo di servizio della coda sono talvolta descritti come eventi OK.

Intervallo servizio coda elevato

Indica che dopo una delle seguenti operazioni:

- Una chiamata MQPUT
- Un'operazione get che lascia una coda non vuota

un'operazione get **non** è stata eseguita in un intervallo di servizi definito dall'utente.

Un'operazione get o una chiamata MQPUT può causare il messaggio di evento Intervallo servizio coda elevato. Gli eventi di valore elevato dell'intervallo di servizio della coda sono talvolta descritti come eventi di valore elevato.

Per abilitare entrambi gli eventi Intervallo servizio coda OK e Intervallo servizio coda Alto, impostare l'attributo di controllo `QServiceIntervalEvent` su Alto. Gli eventi OK intervallo servizio coda vengono abilitati automaticamente quando viene generato un evento Intervallo servizio coda elevato. Non è necessario abilitare gli eventi OK dell'intervallo del servizio coda in modo indipendente.

Gli eventi OK e Alto si escludono a vicenda, quindi se uno è abilitato, l'altro è disabilitato. Tuttavia, entrambi gli eventi possono essere disabilitati contemporaneamente.

Figura 3 a pagina 26 mostra un grafico della profondità della coda rispetto al tempo. All'ora P1, un'applicazione emette un MQPUT per inserire un messaggio nella coda. Alle ore G1, un'altra applicazione emette un MQGET per rimuovere il messaggio dalla coda.

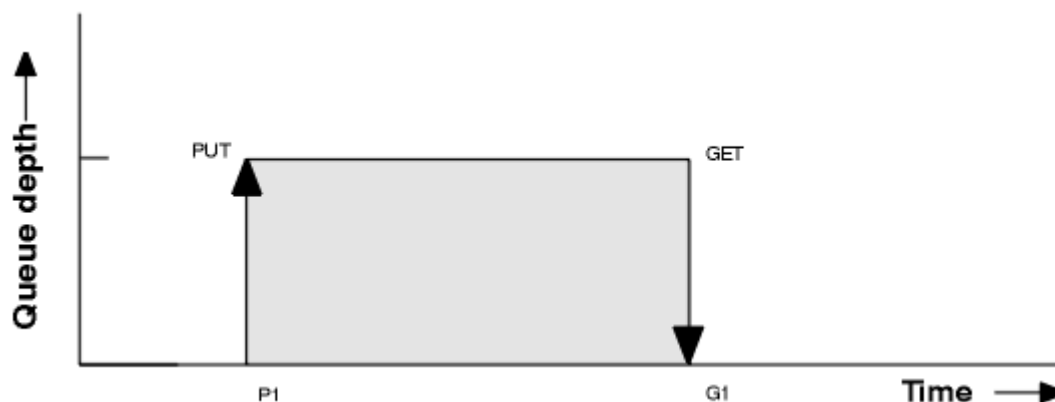


Figura 3. Informazioni sugli eventi dell'intervallo di servizio della coda

I risultati possibili degli eventi dell'intervallo di servizio della coda sono i seguenti:

- Se il tempo trascorso tra l'inserimento e l'ottenimento è minore o uguale all'intervallo di servizio:
 - Un evento *Intervallo servizio coda OK* viene generato alle ore G1, se gli eventi dell'intervallo del servizio coda sono abilitati
- Se il tempo trascorso tra l'inserimento e l'ottenimento è maggiore dell'intervallo di servizio:
 - Un evento *Intervallo servizio coda elevato* viene generato all'ora G1, se gli eventi dell'intervallo del servizio della coda sono abilitati.

L'algoritmo per avviare il timer del servizio e generare gli eventi è descritto in [“Regole per gli eventi dell'intervallo di servizio della coda”](#) a pagina 27.

Riferimenti correlati

[Intervallo servizio coda - OK](#)

[Intervallo servizio coda elevato](#)

[Evento QServiceInterval\(MQLONG\)](#)

[Evento QServiceInterval\(numero intero con segno a 10 cifre\)](#)

Il timer del servizio

Gli eventi dell'intervallo di servizio coda utilizzano un timer interno, denominato *timer servizio*, controllato dal gestore code. Il timer del servizio viene utilizzato solo se è abilitato un evento dell'intervallo del servizio della coda.

Cosa misura esattamente il timer di servizio?

Il timer del servizio misura il tempo trascorso tra una chiamata MQPUT ad una coda vuota o un'operazione get e la successiva operazione put o get, a condizione che la profondità della coda sia diversa da zero tra queste due operazioni.

Quando è attivo il timer di servizio?

Il timer del servizio è sempre attivo (in esecuzione), se la coda contiene messaggi (la profondità è diversa da zero) e un evento dell'intervallo del servizio della coda è abilitato. Se la coda diventa vuota (profondità coda zero), il timer viene impostato su OFF per essere riavviato al successivo inserimento.

Quando viene reimpostato il timer di servizio?

Il timer del servizio viene sempre reimpostato dopo un'operazione get. Viene anche reimpostato da una chiamata MQPUT su una coda vuota. Tuttavia, non viene necessariamente reimpostato su un evento di intervallo di servizio coda.

Come viene utilizzato il timer di servizio?

In seguito a un'operazione get o a una chiamata MQPUT, il gestore code confronta il tempo trascorso misurato dal timer del servizio con l'intervallo di servizio definito dall'utente. Il risultato di questo confronto è che:

- Un evento OK viene generato se è presente un'operazione get e il tempo trascorso è minore o uguale all'intervallo di servizio e questo evento è abilitato.
- Un evento elevato viene generato se il tempo trascorso è maggiore dell'intervallo di servizio e questo evento è abilitato.

Le applicazioni possono leggere il timer di servizio?

No, il timer del servizio è un timer interno che non è disponibile per le applicazioni.

Informazioni sul parametro *TimeSinceReset*

Il parametro *TimeSinceReset* viene restituito come parte delle statistiche evento nei dati evento. Specifica il tempo tra gli eventi successivi dell'intervallo di servizio della coda, a meno che le statistiche evento non vengano reimpostate.

Regole per gli eventi dell'intervallo di servizio della coda

Le regole formali controllano quando il timer del servizio è impostato e vengono generati gli eventi di intervallo del servizio della coda.

Regole per il timer di servizio

Il timer del servizio viene reimpostato su zero e riavviato come segue:

- Dopo una chiamata MQPUT a una coda vuota.
- Dopo una chiamata MQGET, se la coda non è vuota dopo la chiamata MQGET.

La reimpostazione del timer non dipende dalla generazione o meno di un evento.

All'avvio del gestore code, il timer del servizio è impostato sull'ora di avvio se la profondità della coda è maggiore di zero.

Se la coda è vuota dopo un'operazione di acquisizione, il timer viene impostato su OFF.

Eventi di valore elevato intervallo di servizio coda

L'evento Intervallo servizio coda deve essere abilitato (impostato su HIGH).

Gli eventi Intervallo servizio coda elevato vengono automaticamente abilitati quando viene creato un evento Intervallo servizio coda OK.

Se il tempo di servizio è maggiore dell'intervallo di servizio, viene generato un evento o prima della successiva operazione MQPUT o get.

Eventi OK intervallo servizio coda

Gli eventi OK intervallo servizio coda vengono abilitati automaticamente quando viene generato un evento Intervallo servizio coda elevato.

Se il tempo di servizio (tempo trascorso) è inferiore o uguale all'intervallo di servizio, viene generato un evento il o prima della successiva operazione get.

Attività correlate

[“Abilitazione degli eventi di intervallo di servizio della coda” a pagina 28](#)

Per configurare una coda per gli eventi dell'intervallo di servizio della coda, è necessario impostare il gestore code e gli attributi della coda appropriati.

Abilitazione degli eventi di intervallo di servizio della coda

Per configurare una coda per gli eventi dell'intervallo di servizio della coda, è necessario impostare il gestore code e gli attributi della coda appropriati.

Informazioni su questa attività

Gli eventi di valore elevato e OK si escludono a vicenda; ovvero, quando uno è abilitato, l'altro è automaticamente disabilitato:

- Quando un evento elevato viene generato su una coda, il Gestore code disabilita automaticamente gli eventi elevati e abilita gli eventi OK per tale coda.
- Quando un evento OK viene generato su una coda, il gestore code disabilita automaticamente gli eventi OK e abilita eventi elevati per tale coda.

Evento intervallo servizio coda	Attributi Coda
Intervallo servizio coda elevato Intervallo servizio coda - OK Nessun evento di intervallo del servizio coda	QSVCI EV (ALTO) QSVCI EV (OK) QSVCI EV (NONE)
Intervallo di servizio	QSVCI NT (<i>tt</i>) dove <i>tt</i> è il servizio tempo di intervallo in millisecondi.

Effettuare le seguenti operazioni per abilitare gli eventi di intervallo del servizio coda:

Procedura

1. Impostare l'attributo del gestore code **PERFMEV** su **ENABLED**.
Gli eventi delle prestazioni sono abilitati nel gestore code.
2. Impostare l'attributo di controllo, **QSVCI EV**, per un evento Intervallo servizio coda elevato o OK sulla coda, come richiesto.
3. Impostare l'attributo **QSVCI NT** per la coda per specificare l'intervallo di servizio appropriato.

Esempio

Per abilitare gli eventi Intervallo servizio coda elevato con un tempo di servizio di 10 secondi (10 000 millisecondi) utilizzare i seguenti comandi MQSC:

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

Esempi di eventi di intervallo del servizio di coda

Utilizzare gli esempi in questa sezione per comprendere le informazioni che è possibile ottenere dagli eventi dell'intervallo di servizio della coda.

I tre esempi di sottoargomento forniscono illustrazioni progressivamente più complesse dell'utilizzo degli eventi dell'intervallo di servizio della coda.

Le cifre che accompagnano gli esempi in ogni sottoargomento hanno la stessa struttura:

- La Figura 1 è un grafico della profondità della coda rispetto al tempo, che mostra le singole chiamate MQGET e MQPUT.
- La sezione Commenti mostra un confronto dei vincoli temporali. Ci sono tre periodi di tempo che è necessario considerare:
 - L'intervallo di servizio definito dall'utente.
 - Il tempo misurato dal timer del servizio.
 - L'ora dell'ultima reimpostazione delle statistiche evento (TimeSinceReimpostazione nei dati evento).
- La sezione Riepilogo statistiche evento mostra quali eventi sono abilitati in qualsiasi momento e quali eventi vengono generati.

Gli esempi illustrano i seguenti aspetti degli eventi dell'intervallo di servizio della coda:

- Il modo in cui la profondità della coda varia nel tempo.
- Il confronto tra il tempo trascorso misurato dal timer del servizio e l'intervallo del servizio.
- Quale evento è abilitato.
- Quali eventi vengono generati.

Attenzione: L'esempio 1 mostra un caso semplice in cui i messaggi sono intermittenti e ciascun messaggio viene rimosso dalla coda prima dell'arrivo del messaggio successivo. Dai dati evento, si sa che il numero massimo di messaggi sulla coda era uno. È quindi possibile determinare per quanto tempo ciascun messaggio è stato in coda.

Tuttavia, nel caso generale, in cui è presente più di un messaggio nella coda e la sequenza di chiamate MQGET e MQPUT non è prevedibile, non è possibile utilizzare gli eventi di intervallo di servizio della coda per calcolare per quanto tempo un singolo messaggio rimane su una coda. Il parametro di reimpostazione TimeSince, restituito nei dati evento, può includere una proporzione di tempo in cui non sono presenti messaggi nella coda. Pertanto, tutti i risultati che si ricavano da queste statistiche sono implicitamente mediati per includere questi tempi.

Concetti correlati

“Eventi intervallo servizio coda” a pagina 26

Gli eventi dell'intervallo di servizio della coda indicano se un'operazione è stata eseguita su una coda all'interno di un intervallo di tempo definito dall'utente denominato *intervallo di servizio*. A seconda dell'installazione, è possibile utilizzare gli eventi dell'intervallo di servizio della coda per monitorare se i messaggi vengono rimossi dalle code abbastanza rapidamente.

“Il timer del servizio” a pagina 27

Gli eventi dell'intervallo di servizio coda utilizzano un timer interno, denominato *timer servizio*, controllato dal gestore code. Il timer del servizio viene utilizzato solo se è abilitato un evento dell'intervallo del servizio della coda.

Eventi intervallo servizio coda: esempio 1

Una sequenza di base di chiamate MQGET e MQPUT, in cui la profondità della coda è sempre uno o zero.

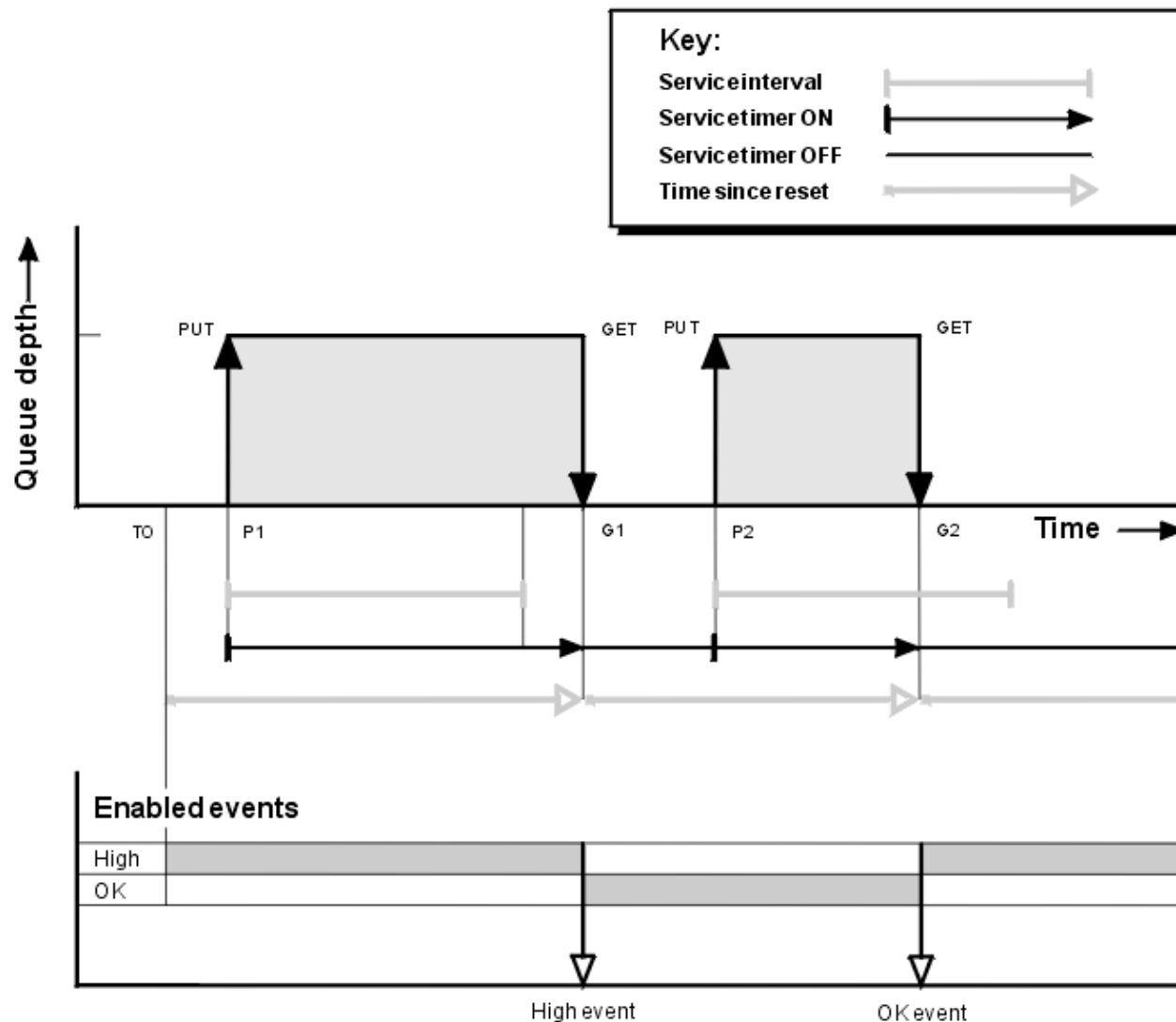


Figura 4. Eventi intervallo servizio coda - esempio 1

Commento

1. In P1, un'applicazione inserisce un messaggio in una coda vuota. Questo avvia il timer del servizio.
Si noti che T0 potrebbe essere l'ora di avvio del gestore code.
2. Su G1, un'altra applicazione riceve il messaggio dalla coda. Poiché il tempo trascorso tra P1 e G1 è maggiore dell'intervallo di servizio, viene generato un evento Intervallo servizio coda elevato sulla

chiamata MQGET in G1. Quando l'evento elevato viene generato, il gestore code reimposta l'attributo di controllo eventi in modo che:

- a. L'evento OK viene abilitato automaticamente.
- b. L'evento alto è disabilitato.

Poiché la coda è ora vuota, il temporizzatore di servizio viene impostato su OFF.

3. In P2, un secondo messaggio viene inserito nella coda. Questo riavvia il timer del servizio.
4. In G2, il messaggio viene rimosso dalla coda. Tuttavia, poiché il tempo trascorso tra P2 e G2 è inferiore all'intervallo di servizio, un evento Intervallo servizio coda OK viene generato sulla chiamata MQGET in G2. Quando viene generato l'evento OK, il gestore code reimposta l'attributo di controllo in modo che:
 - a. L'evento alto viene abilitato automaticamente.
 - b. L'evento OK è disabilitato.

Poiché la coda è vuota, il temporizzatore di servizio viene nuovamente commutato in uno stato OFF.

Riepilogo statistiche evento

Tabella 5 a pagina 31 riepiloga le statistiche evento per questo esempio.

<i>Tabella 5. Riepilogo statistiche evento, ad esempio 1</i>		
Attributo	Evento 1	Evento 2
Ora dell'evento	T (G1)	T (G2)
Il tipo di evento	Elevato	OK
TimeSinceReset	T (G1) - T (0)	T (G2) - T (G1)
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

La parte centrale di [Figura 4 a pagina 30](#) mostra il tempo trascorso misurato dal timer del servizio rispetto all'intervallo di servizio per tale coda. Per vedere se potrebbe verificarsi un evento di intervallo di servizio della coda, confrontare la lunghezza della linea orizzontale che rappresenta il timer del servizio (con freccia) con quella della linea che rappresenta l'intervallo di servizio. Se la linea del timer del servizio è più lunga e l'evento Intervallo servizio coda elevato è abilitato, un evento Intervallo servizio coda elevato si verifica al successivo richiamo. Se la linea del timer è più corta e l'evento Intervallo servizio coda OK è abilitato, un evento Intervallo servizio coda OK si verifica al successivo richiamo.

Eventi intervallo servizio coda: esempio 2

Una sequenza di chiamate MQPUT e MQGET, dove la profondità della coda non è sempre uno o zero.

Questo esempio mostra anche le istanze del timer da reimpostare senza che vengano generati eventi, ad esempio, all'ora P2.

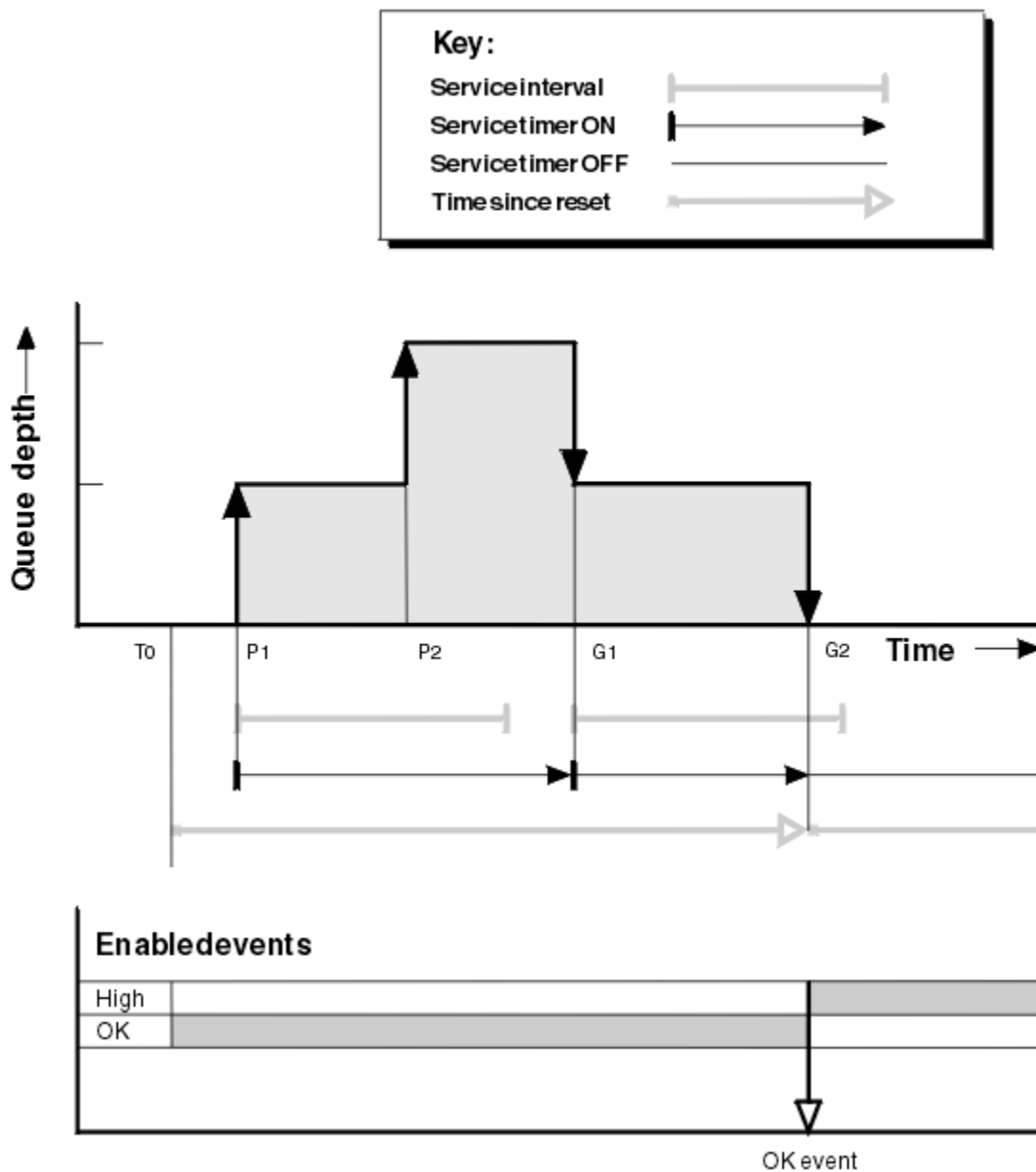


Figura 5. Eventi intervallo servizio coda - esempio 2

Commento

In questo esempio, gli eventi OK sono abilitati inizialmente e le statistiche della coda sono state reimpostate al momento T0.

1. In P1, il primo inserimento avvia il temporizzatore di servizio.
2. In P2, il secondo inserimento non genera un evento perché un inserimento non può causare un evento OK.
3. A G1, l'intervallo di servizio è stato ora superato e quindi non viene generato un evento OK. Tuttavia, la chiamata MQGET provoca la reimpostazione del timer del servizio.
4. In G2, il secondo richiamo si verifica nell'intervallo di servizio e questa volta viene generato un evento OK. Il gestore code reimposta l'attributo di controllo eventi in modo che:
 - a. L'evento alto viene abilitato automaticamente.
 - b. L'evento OK è disabilitato.

Poiché la coda è ora vuota, il temporizzatore di servizio viene impostato su OFF.

Riepilogo statistiche evento

Tabella 6 a pagina 33 riepiloga le statistiche evento per questo esempio.

<i>Tabella 6. Riepilogo statistiche evento, ad esempio 2</i>	
Attributo	Evento 2
Ora dell'evento	T (G2)
Il tipo di evento	OK
TimeSinceReset	T (G2) - T (0)
HighQDepth	2
MsgEnqCount	2
MsgDeqCount	2

Eventi intervallo servizio coda: esempio 3

Una sequenza di chiamate MQGET e MQPUT più sporadica rispetto agli esempi precedenti.

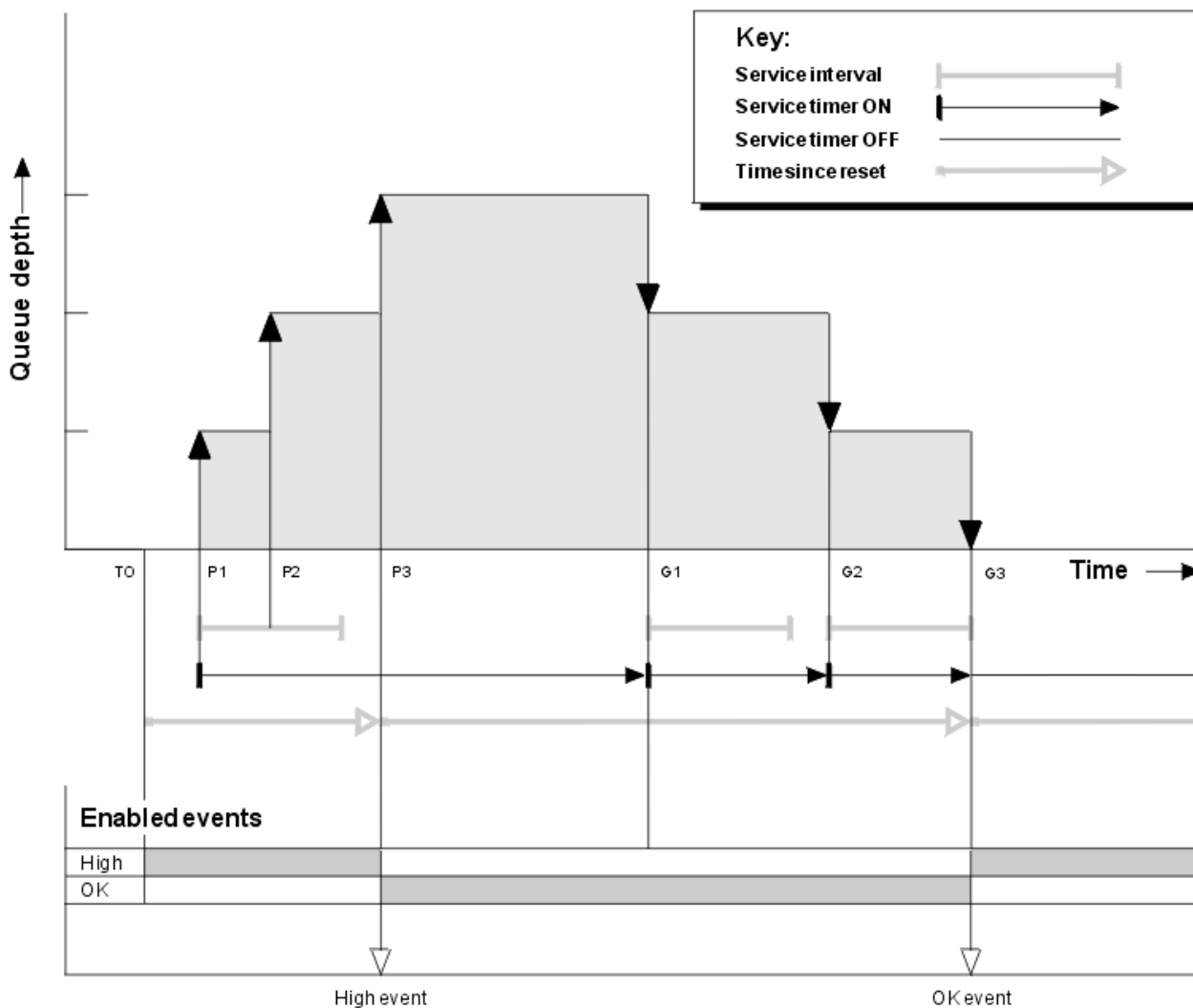


Figura 6. Eventi intervallo servizio coda - esempio 3

Commento

1. Al tempo T (0), le statistiche della coda vengono reimpostate e gli eventi Intervallo di servizio coda elevato sono abilitati.
2. In P1, il primo inserimento avvia il temporizzatore di servizio.
3. In P2, il secondo inserimento aumenta la profondità della coda a due. Un evento elevato non viene generato qui perché l'intervallo di tempo del servizio non è stato superato.
4. In P3, il terzo inserimento genera un evento elevato. (Il timer ha superato l'intervallo di servizio.) Il temporizzatore non viene reimpostato perché la profondità della coda non era zero prima dell'operazione di inserimento. Tuttavia, gli eventi OK sono abilitati.
5. In G1, la chiamata MQGET non genera un evento perché l'intervallo di servizio è stato superato e gli eventi OK sono abilitati. La chiamata MQGET, tuttavia, reimposta il timer del servizio.
6. In G2, la chiamata MQGET non genera un evento perché l'intervallo del servizio è stato superato e gli eventi OK sono abilitati. Di nuovo, la chiamata MQGET reimposta il timer del servizio.
7. In G3, il terzo get svuota la coda e il timer del servizio è uguale all'intervallo del servizio. Pertanto, viene creato un evento OK. Il timer del servizio viene reimpostato e gli eventi elevati sono abilitati. La chiamata MQGET svuota la coda e il timer viene impostato sullo stato OFF.

Riepilogo statistiche evento

Tabella 7 a pagina 34 riepiloga le statistiche evento per questo esempio.

<i>Tabella 7. Riepilogo statistiche evento, ad esempio 3</i>		
Attributo	Evento 1	Evento 2
Ora dell'evento	T (P3)	T (G3)
Il tipo di evento	Elevato	OK
TimeSinceReset	T (P3) - T (0)	T (G3) - T (P3)
HighQDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

Eventi di profondità della coda

Gli eventi di profondità della coda sono correlati alla profondità della coda, ovvero al numero di messaggi nella coda.

Nelle applicazioni IBM MQ, le code non devono diventare piene. In tal caso, le applicazioni non possono più inserire i messaggi nella coda specificata. Anche se il messaggio non viene perso se ciò si verifica, una coda completa può causare notevoli disagi. Il numero di messaggi può essere accumulato su una coda se i messaggi vengono inseriti nella coda più velocemente di quanto le applicazioni che li elaborano possano disattivarli.

La soluzione a questo problema dipende dalle circostanze particolari, ma potrebbe comportare:

- Deviazione di alcuni messaggi in un'altra coda.
- Avvio di nuove applicazioni per togliere più messaggi dalla coda.
- Arresto del traffico di messaggi non essenziali.
- Aumento della profondità della coda per superare un valore massimo transitorio.

L'avvertenza preventiva che i problemi potrebbero essere in corso rende più facile l'azione preventiva. A tale scopo, IBM MQ fornisce i seguenti eventi di profondità della coda:

Eventi coda profondità massima

Indica che la profondità della coda è stata aumentata a una soglia predefinita denominata Limite massimo di profondità della coda.

Eventi profondità coda minima

Indica che la profondità della coda è diminuita a una soglia predefinita denominata Limite minimo profondità coda.

Eventi coda piena

Indica che la coda ha raggiunto la sua massima profondità, ovvero che la coda è piena.

Un evento Coda completa viene generato quando un'applicazione tenta di inserire un messaggio su una coda che ha raggiunto la profondità massima. Gli eventi Grandezza coda elevata forniscono un'avvertenza in anticipo che una coda si sta riempiendo. Ciò significa che, dopo aver ricevuto questo evento, l'amministratore di sistema deve eseguire alcune azioni preventive. È possibile configurare il gestore code in modo che, se l'azione preventiva ha esito positivo e la profondità della coda scende a un livello più sicuro, il gestore code genera un evento Profondità minima coda.

L'esempio del primo evento di profondità della coda illustra l'effetto di un'azione presunta che impedisce il riempimento della coda.

Concetti correlati

[“Esempi di eventi di profondità della coda” a pagina 38](#)

Utilizzare questi esempi per comprendere le informazioni che è possibile ottenere dagli eventi di profondità della coda

Riferimenti correlati

[Coda piena](#)

[Grandezza coda elevata](#)



[Grandezza coda bassa](#)

Abilitazione degli eventi di profondità della coda

Per configurare una coda per qualsiasi evento di profondità della coda, impostare il gestore code e gli attributi della coda appropriati.

Informazioni su questa attività

Per impostazione predefinita, tutti gli eventi di profondità della coda sono disabilitati. Quando questa opzione è abilitata, gli eventi di profondità della coda vengono generati come segue:

- Un evento Grandezza coda elevata viene generato quando un messaggio viene inserito nella coda, causando una grandezza della coda maggiore o uguale al valore impostato per **QDepthHighLimit**.
 - Un evento Grandezza coda elevata viene abilitato automaticamente da un evento Grandezza coda bassa sulla stessa coda.
 - Un evento Grandezza coda elevata abilita automaticamente un evento Grandezza coda bassa e un evento Coda piena sulla stessa coda.
-  **Multi** Un evento Grandezza coda bassa viene generato quando un messaggio viene rimosso da una coda da un'operazione GET, causando una profondità della coda inferiore o uguale al valore impostato per **QDepthLowLimit**.
-  **z/OS** Un evento Profondità minima della coda viene generato quando un messaggio viene rimosso da una coda da un'operazione GET o sarebbe stato rimosso ma è scaduto, causando una profondità della coda inferiore o uguale al valore impostato per **QDepthLowLimit**.
 - Un evento Grandezza coda bassa viene abilitato automaticamente da un evento Grandezza coda alta o da un evento Coda piena sulla stessa coda.
 - Un evento Profondità coda bassa abilita automaticamente un evento Profondità coda alta e un evento Coda piena sulla stessa coda.
- Un evento Coda completa viene generato quando un'applicazione non è in grado di inserire un messaggio in una coda perché la coda è piena.
 - Un evento Coda piena viene abilitato automaticamente da un evento Grandezza coda elevata o Grandezza coda bassa sulla stessa coda.

– Un evento Coda piena abilita automaticamente un evento Grandezza coda bassa sulla stessa coda.
Per configurare una coda per uno degli eventi di profondità della coda, effettuare le seguenti operazioni:

Procedura

1. Abilitare gli eventi delle prestazioni sul gestore code, utilizzando l'attributo gestore code **PERFMEV**.
Gli eventi vanno alla coda SYSTEM.ADMIN.PERFM.EVENT.
2. Impostare uno dei seguenti attributi per abilitare l'evento sulla coda richiesta:
 - Evento *QDepthHigh* (**QDPHIEV** in MQSC)
 - *QDepthLowEvent* (**QDPLOEV** in MQSC)
 - *QDepthMaxEvent* (**QDPMAXEV** in MQSC)
3. Opzionale: Per impostare i limiti, assegnare i seguenti attributi, come percentuale della profondità massima della coda:
 - *QDepthHighQDepthHigh* (**QDEPTHHI** in MQSC)
 - *QDepthLowQDepthLow* (**QDEPTHLO** in MQSC)

Limitazione: **QDEPTHHI** non deve essere minore di **QDEPTHLO**.

Se **QDEPTHHI** è uguale a **QDEPTHLO**, viene generato un messaggio di evento ogni volta che la profondità della coda passa il valore in entrambe le direzioni, poiché la soglia superiore è abilitata quando la profondità della coda è inferiore al valore e la soglia inferiore è abilitata quando la profondità è superiore al valore.

Risultati

Nota:

Multi Un evento Profondità minima della coda non viene generato quando i messaggi scaduti vengono rimossi da una coda da un'operazione GET che fa sì che la profondità della coda sia inferiore o uguale al valore impostato per **QDepthLowLimit**. IBM MQ genera il messaggio di evento basso della profondità della coda solo durante un'operazione GET riuscita. Di conseguenza, quando i messaggi scaduti vengono rimossi dalla coda, non viene generato alcun messaggio di evento di bassa profondità della coda. Inoltre, dopo la rimozione di questi messaggi scaduti dalla coda, *QDepthHighEvent* e *QDepthLowEvent* non vengono reimpostate.

z/OS IBM MQ genera il messaggio di evento di bassa profondità della coda durante un'operazione GET distruttiva riuscita o un'operazione GET distruttiva che avrebbe avuto esito positivo se un messaggio corrispondente non fosse scaduto. In alternativa, quando i messaggi scaduti vengono rimossi da una coda durante l'elaborazione in background generale, non viene generato alcun messaggio di evento di profondità della coda bassa. Inoltre, dopo la rimozione dei messaggi scaduti da una coda durante l'elaborazione in background generale, *QDepthHighEvent* e *QDepthLowEvent* non vengono reimpostati. Per ulteriori informazioni sull'elaborazione dei messaggi scaduti, consultare [Ottimizzazione del gestore code su IBM MQ for z/OS](#).

Esempio

Per abilitare gli eventi Grandezza coda elevata sulla coda MYQUEUE con un limite impostato su 80%, utilizzare i seguenti comandi MQSC:

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

Per abilitare gli eventi Grandezza coda bassa sulla coda MYQUEUE con un limite impostato al 20%, utilizzare i seguenti comandi MQSC:

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

Per abilitare gli eventi Coda piena sulla coda MYQUEUE, utilizzare i seguenti comandi MQSC:

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

▶ z/OS *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.

Esempi di eventi di profondità della coda

Utilizzare questi esempi per comprendere le informazioni che è possibile ottenere dagli eventi di profondità della coda

Il primo esempio fornisce un'illustrazione di base degli eventi di profondità della coda. Il secondo esempio è più ampio, ma i principi sono gli stessi del primo. Entrambi gli esempi utilizzano la stessa definizione di coda, come segue:

La coda, MYQUEUE1, ha una profondità massima di 1000 messaggi. Il limite massimo di profondità della coda è 80% e il limite minimo di profondità della coda è 20%. Inizialmente, gli eventi Grandezza coda elevata sono abilitati, mentre gli altri eventi Grandezza coda sono disabilitati.

I comandi IBM MQ (MQSC) per configurare questa coda sono:

```
ALTER QMGR PERFMEV(ENABLED)

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

Concetti correlati

[“Eventi di profondità della coda” a pagina 34](#)

Gli eventi di profondità della coda sono correlati alla profondità della coda, ovvero al numero di messaggi nella coda.

Attività correlate

[“Abilitazione degli eventi di profondità della coda” a pagina 35](#)

Per configurare una coda per qualsiasi evento di profondità della coda, impostare il gestore code e gli attributi della coda appropriati.

Riferimenti correlati

[Comandi MQSC](#)

Eventi di profondità della coda: esempio 1

Una sequenza di base di eventi di profondità della coda.

Figura 7 a pagina 39 mostra la variazione della profondità della coda nel tempo.

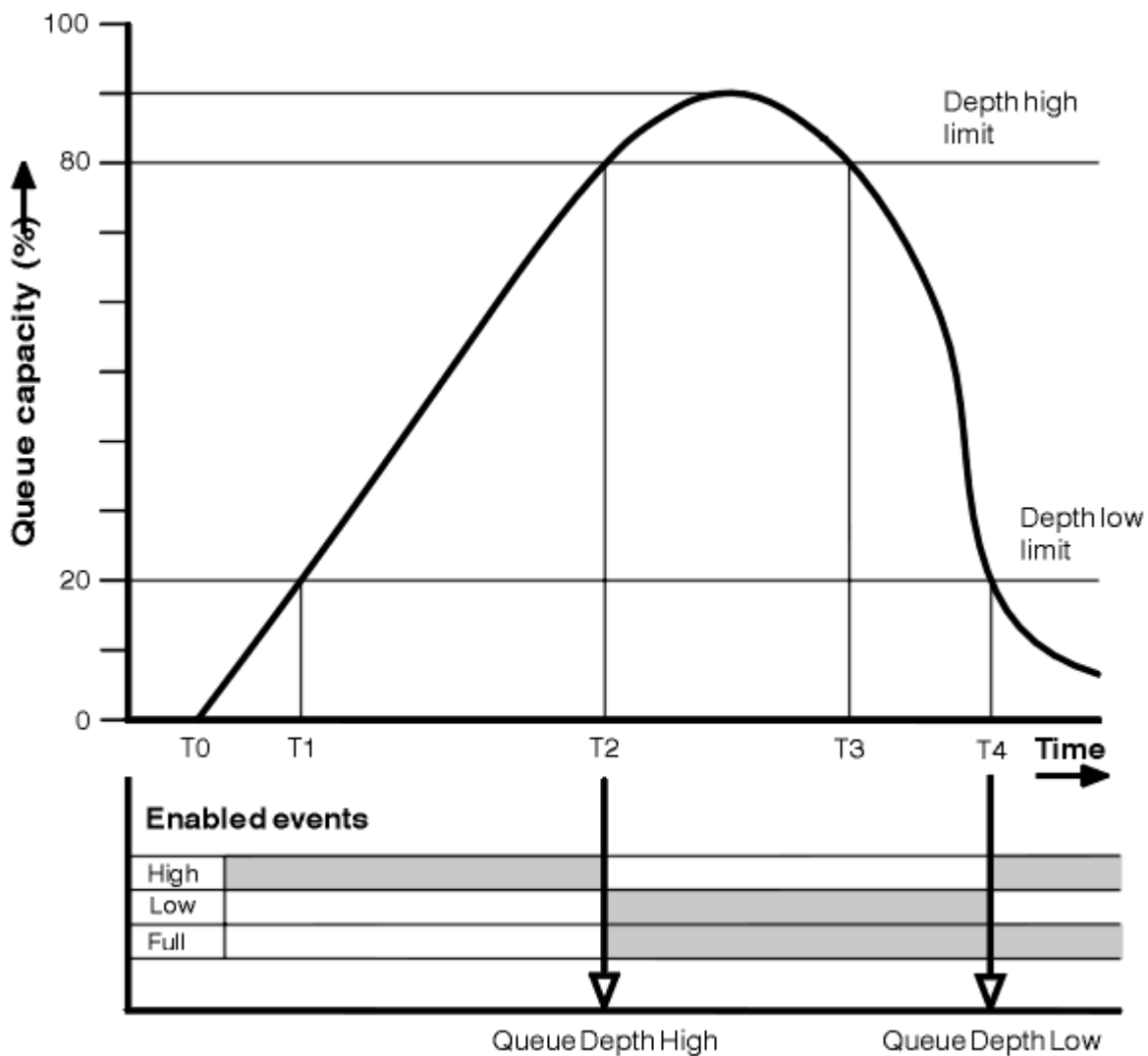


Figura 7. Eventi di profondità coda (1)

Commento

1. A T (1), la profondità della coda è in aumento (più chiamate MQPUT che chiamate MQGET) e supera il limite inferiore di profondità della coda. In questo momento non viene generato alcun evento.
Ciò abilita gli eventi Grandezza coda e Grandezza coda bassa.
2. La profondità della coda continua ad aumentare fino a T (2), quando viene raggiunto il limite massimo di profondità (80%) e viene generato un evento Grandezza coda elevata.
Ciò abilita gli eventi Grandezza coda e Grandezza coda bassa.
3. Le azioni preventive (presunte) istigate dall'evento impediscono alla coda di diventare piena. Con il tempo T (3), il limite di profondità massima della coda è stato nuovamente raggiunto, questa volta dall'alto. In questo momento non viene generato alcun evento.
4. La profondità della coda continua a diminuire fino a T (4), quando raggiunge il limite minimo di profondità (20%) e viene generato un evento Profondità minima coda.
Ciò abilita gli eventi Coda piena e Grandezza coda elevata.

Riepilogo statistiche evento

Tabella 8 a pagina 40 riepiloga le statistiche degli eventi coda e Tabella 9 a pagina 40 riepiloga quali eventi sono abilitati.

Tabella 8. Riepilogo statistiche evento per gli eventi di profondità della coda (esempio 1)

Statistiche di eventi	Evento 2	Evento 4
Ora dell'evento	T (2)	T (4)
Il tipo di evento	Grandezza coda elevata	Grandezza coda bassa
TimeSinceReset	T (2) - T (0)	T (4) - T (2)
HighQDepth (Grandezza massima coda dalla reimpostazione)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

Tabella 9. Riepilogo che mostra quali eventi sono abilitati

Periodo di tempo	evento Grandezza coda elevata	Evento Grandezza coda bassa	evento Coda piena
Prima di T (1)	Abilitato	-	-
Da T (1) a T (2)	Abilitato	-	-
Da T (2) a T (3)	-	Abilitato	Abilitato
Da T (3) a T (4)	-	Abilitato	Abilitato
Dopo T (4)	Abilitato	-	Abilitato

Eventi di profondità della coda: esempio 2

Una sequenza più estesa di eventi di profondità della coda.

Figura 8 a pagina 41 mostra la variazione della profondità della coda nel tempo.

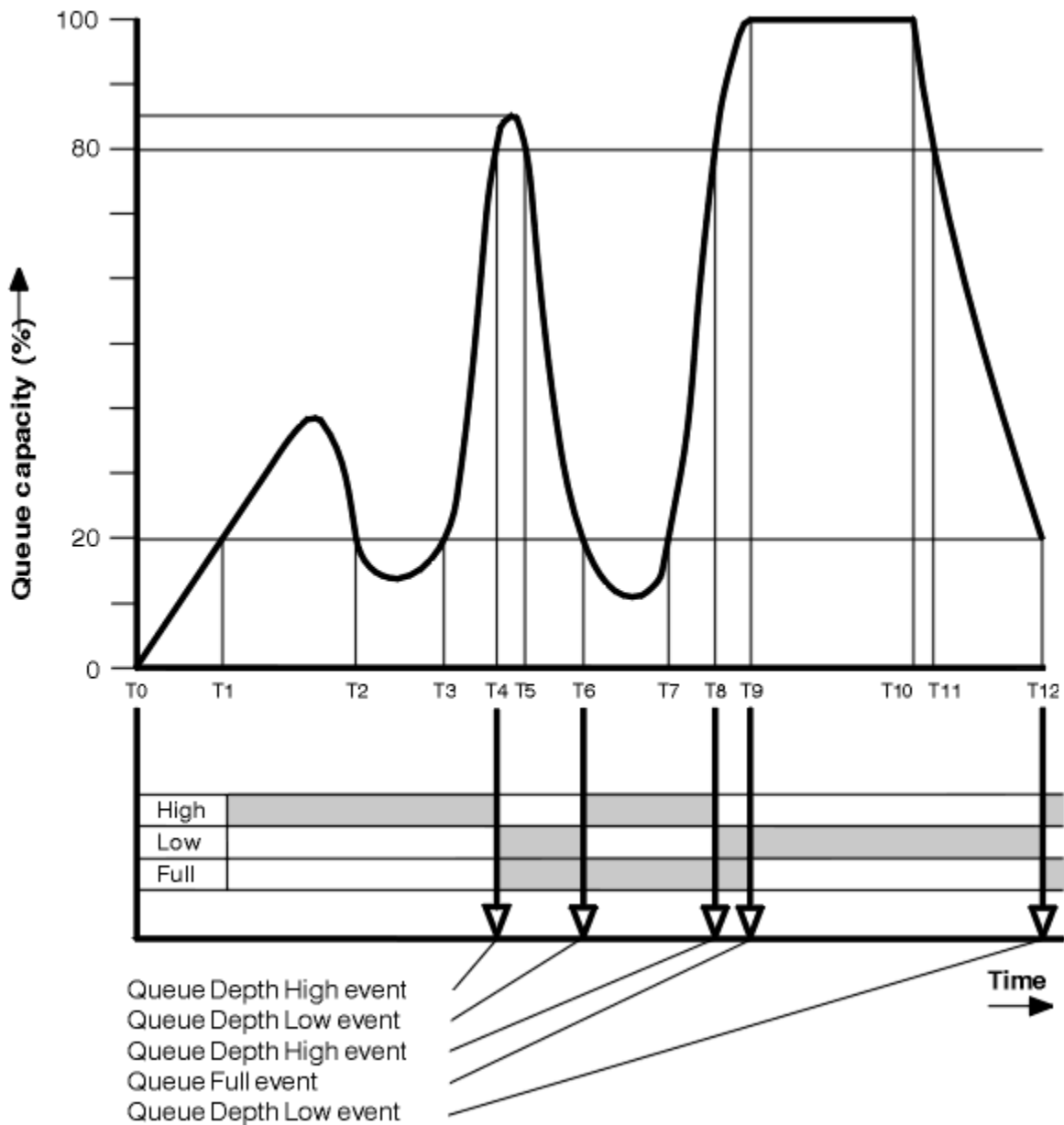


Figura 8. Eventi di profondità coda (2)

Commento

- Non viene generato alcun evento Profondità minima coda nei seguenti orari:
 - T (1) (La profondità della coda aumenta e non è abilitata)
 - T (2) (Non abilitato)
 - T (3) (La profondità della coda aumenta e non è abilitata)
- Alle T (4) si verifica un evento Grandezza coda elevata. Ciò abilita gli eventi Grandezza coda e Grandezza coda bassa.
- Alle T (9) si verifica un evento Coda piena **dopo** il primo messaggio che non può essere inserito nella coda poiché la coda è piena.
- Al punto T (12) si verifica un evento Profondità coda bassa.

Riepilogo statistiche evento

Tabella 10 a pagina 42 riepiloga le statistiche degli eventi della coda e Tabella 11 a pagina 42 riepiloga gli eventi abilitati in momenti diversi per questo esempio.

Statistiche di eventi	Evento 4	Evento 6	Evento 8	Evento 9	Evento 12
Ora dell'evento	T (4)	T (6)	T (8)	T (9)	T (12)
Il tipo di evento	Grandezza coda elevata	Grandezza coda bassa	Grandezza coda elevata	Coda piena	Grandezza coda bassa
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

Periodo di tempo	evento Grandezza coda elevata	Evento Grandezza coda bassa	evento Coda piena
Da T (0) a T (4)	Abilitato	-	-
Da T (4) a T (6)	-	Abilitato	Abilitato
Da T (6) a T (8)	Abilitato	-	Abilitato
Da T (8) a T (9)	-	Abilitato	Abilitato
Da T (9) a T (12)	-	Abilitato	-
Dopo T (12)	Abilitato	-	Abilitato

Nota: Gli eventi non sono nel punto di sincronizzazione. Pertanto è possibile avere una coda vuota, quindi riempirla causando un evento, quindi eseguire il rollback di tutti i messaggi sotto il controllo di un gestore del punto di sincronizzazione. Tuttavia, l'abilitazione degli eventi è stata impostata automaticamente, in modo che la volta successiva in cui la coda si riempie, non viene generato alcun evento.

Eventi di configurazione

Gli eventi di configurazione sono notifiche generate quando un oggetto viene creato, modificato o eliminato e possono essere generate anche da richieste esplicite.

Consultare [“Controllo degli eventi di configurazione, comando e programma di registrazione”](#) a pagina 22 per informazioni su come attivare gli eventi.

Gli eventi di configurazione notificano le modifiche apportate agli attributi di un oggetto. Esistono quattro tipi di eventi di configurazione:

- Crea eventi oggetto
- Eventi di modifica oggetto
- Elimina eventi oggetto
- Aggiorna eventi oggetto

I dati evento contengono le seguenti informazioni:

Informazioni sull'origine

comprende il gestore code da cui è stata apportata la modifica, l'ID dell'utente che ha apportato la modifica e il modo in cui è stata apportata la modifica, ad esempio mediante un comando della console.

Informazioni contesto

una replica delle informazioni di contesto nei dati del messaggio dal messaggio di comandi.

Le informazioni di contesto sono incluse nei dati evento solo quando il comando è stato immesso come messaggio sul SISTEMA.SYSTEM.COMMAND.INPUT .

Identità oggetto

comprende il nome, il tipo e la disposizione dell'oggetto.

Attributi oggetto

comprende i valori di tutti gli attributi nell'oggetto.

L'evento è un messaggio in formato PCF. Per ulteriori informazioni, consultare:

- [Modifica oggetto](#)
- [Crea oggetto](#)
- [Elimina oggetto](#)
- [Aggiorna oggetto](#)

Nel caso di eventi oggetto di modifica, vengono prodotti un messaggio precedente e un messaggio successivo. Hanno i seguenti campi in comune, in modo da poter identificare i messaggi corrispondenti:

- Lo stesso ID di correlazione
- Nell'intestazione PCF:
 - L'intestazione del messaggio precedente ha `MsgSeqNumber' : 1, 'Control' : 'NOT_LAST'`
 - L'intestazione del messaggio successivo ha `MsgSeqNumber' : 2, 'Control' : 'LAST'`

Ogni messaggio di evento di configurazione generato viene inserito nella coda SYSTEM.ADMIN.CONFIG.EVENT.

Concetti correlati

[“Eventi di configurazione” a pagina 17](#)

Gli eventi di configurazione vengono generati quando un evento di configurazione viene richiesto esplicitamente o automaticamente quando un oggetto viene creato, modificato o eliminato.

Riferimenti correlati

[Crea oggetto](#)

[Modifica oggetto](#)

[Elimina oggetto](#)

[Aggiorna oggetto](#)

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

Generazione evento di configurazione

Utilizzare questa pagina per visualizzare i comandi che causano la generazione di eventi di configurazione e per comprendere le circostanze in cui gli eventi di configurazione non vengono generati

Un messaggio di evento di configurazione viene inserito nella coda degli eventi di configurazione quando l'attributo del gestore code CONFIGEV è ENABLED e

- viene immesso uno dei seguenti comandi o il loro equivalente PCF:
 - DELETE AUTINFO
 - DELETE CFSTRUCT
 - Elimina canale

- Eliminazione elenco nomi
- Eliminazione processo
- ELIMINARE QMODEL/QALIAS/QREMOTE
- DELETE STGCLASS
- Elimina argomento
- AGGIORNA QMGR
- i seguenti comandi, o il loro equivalente PCF, vengono emessi anche se non vi è alcuna modifica all'oggetto:
 - DEFINE/ALTER AUTINFO
 - DEFINE/ALTER CFSTRUCT
 - DEFINIZIONE/MODIFICA CANALE
 - ELENCO NOMI DEFINIZIONE / ALTER
 - DEFINE/ALTER PROCESSO
 - DEFINE/ALTER QMODEL/QALIAS/QREMOTE
 - DEFINE/ALTER STGCLASS
 - DEFINIZIONE/MODIFICA ARGOMENTO
 - DEFINE MAXSMSGS
 - SET CHLAUTH
 - ALTER QMGR, a meno che l'attributo CONFIGEV non sia DISABLED e non sia modificato in ENABLED
- uno dei seguenti comandi, o il loro equivalente PCF, viene emesso per una coda locale che non è dinamica temporanea, anche se non vi è alcuna modifica alla coda.
 - ELIMINA QLOCALE
 - QLOCALE DEFINE/ALTER
- viene emessa una chiamata MQSET, diversa da quella per una coda dinamica temporanea, anche se non vi è alcuna modifica all'oggetto.

Quando non vengono generati eventi di configurazione

I messaggi degli eventi di configurazione non vengono generati nelle circostanze seguenti:

- Quando un comando o una chiamata MQSET ha esito negativo
- Quando un gestore code rileva un errore nel tentativo di inserire un evento di configurazione nella coda eventi, nel qual caso il comando o la chiamata MQSET vengono completati, ma non viene generato alcun messaggio di evento
- Per una coda dinamica temporanea
- Quando vengono apportate modifiche interne all'attributo della coda TRIGGER
- Per la coda eventi di configurazione SYSTEM.ADMIN.CONFIG.EVENT, ad eccezione del comando REFRESH QMGR
- Per i comandi REFRESH/RESET CLUSTER e RESUME/SUSPEND QMGR che causano modifiche del cluster
- Durante la creazione o l'eliminazione di un gestore code

Concetti correlati

Introduzione ai formati di comando programmabili

“Eventi di configurazione” a pagina 42

Gli eventi di configurazione sono notifiche generate quando un oggetto viene creato, modificato o eliminato e possono essere generate anche da richieste esplicite.

Riferimenti correlati

Comandi MQSC

[MQSET - Imposta attributi oggetto](#)

[MQSET - Imposta attributi oggetto](#)

Utilizzo evento di configurazione

Utilizzare questa pagina per visualizzare in che modo è possibile utilizzare gli eventi di configurazione per ottenere informazioni sul sistema e per comprendere i fattori, come CMDSCOPE, che possono influire sull'utilizzo degli eventi di configurazione.

È possibile utilizzare gli eventi di configurazione per i seguenti scopi:

1. Per produrre e gestire un repository di configurazione centrale, da cui è possibile produrre report e generare informazioni sulla struttura del sistema.
2. Per creare una traccia di controllo. Ad esempio, se un oggetto viene modificato in modo imprevisto, è possibile memorizzare le informazioni relative a chi ha apportato la modifica e quando è stata eseguita.

Ciò può essere particolarmente utile quando sono abilitati anche gli eventi comando. Se un comando MQSC o PCF genera un evento di configurazione e un evento di comando, entrambi i messaggi di evento condivideranno lo stesso identificativo di correlazione nel relativo descrittore del messaggio.

Per una chiamata MQSET o uno dei seguenti comandi:

- oggetto DEFINE
- Oggetto ALTER
- Elimina oggetto

se l'attributo CONFIGEV del gestore code è abilitato, ma il messaggio dell'evento di configurazione non può essere inserito nella coda degli eventi di configurazione, ad esempio la coda degli eventi non è stata definita, il comando o la chiamata MQSET vengono eseguiti indipendentemente.

Effetti di CMDSCOPE

Per i comandi in cui viene utilizzato CMDSCOPE, il messaggio o i messaggi dell'evento di configurazione verranno generati sul gestore code o sui gestori code in cui viene eseguito il comando, non dove viene immesso il comando. Tuttavia, tutte le informazioni di origine e di contesto nei dati evento saranno correlate al comando originale immesso, anche se il comando che utilizza CMDSCOPE è stato generato dal gestore code di origine.

Se un gruppo di condivisione code include gestori code che non sono alla versione corrente, gli eventi verranno generati per qualsiasi comando eseguito tramite CMDSCOPE su un gestore code che è alla versione corrente, ma non su quelli che sono alla versione precedente. Ciò si verifica anche se il gestore code in cui viene immesso il comando è alla versione precedente, anche se in tal caso non vengono incluse informazioni di contesto nei dati evento.

Concetti correlati

[Introduzione ai formati di comando programmabili](#)

[“Eventi di configurazione” a pagina 42](#)

Gli eventi di configurazione sono notifiche generate quando un oggetto viene creato, modificato o eliminato e possono essere generate anche da richieste esplicite.

Riferimenti correlati

[MQSET - Imposta attributi oggetto](#)

[MQSET - Imposta attributi oggetto](#)

Evento Aggiorna configurazione oggetto

L'evento di configurazione Aggiorna oggetto è diverso dagli altri eventi di configurazione, poiché si verifica solo quando esplicitamente richiesto.

Gli eventi di creazione, modifica ed eliminazione vengono generati da una chiamata MQSET o da un comando per modificare un oggetto, ma l'evento di aggiornamento dell'oggetto si verifica solo quando esplicitamente richiesto dal comando MQSC, REFRESH QMGR o dall'equivalente PCF.

Il comando REFRESH QMGR è diverso da tutti gli altri comandi che generano eventi di configurazione. Tutti gli altri comandi si applicano a un particolare oggetto e generano un singolo evento di configurazione per tale oggetto. Il comando REFRESH QMGR può produrre molti messaggi di eventi di configurazione che potenzialmente rappresentano ogni definizione di oggetto memorizzata da un gestore code. Viene generato un messaggio evento per ogni oggetto selezionato.

Il comando REFRESH QMGR utilizza una combinazione di tre criteri di selezione per filtrare il numero di oggetti coinvolti:

- Nome oggetto
- Tipo oggetto
- Intervallo di aggiornamento

Se non si specifica alcun criterio di selezione nel comando REFRESH QMGR, vengono utilizzati i valori predefiniti per ciascun criterio di selezione e viene generato un messaggio di evento di configurazione aggiornamento per ogni definizione di oggetto memorizzata dal gestore code. Ciò potrebbe causare tempi di elaborazione non accettabili e la generazione di messaggi di evento. Specificare alcuni criteri di selezione.

Il comando REFRESH QMGR che genera gli eventi di aggiornamento può essere utilizzato nelle seguenti situazioni:

- Quando si desiderano i dati di configurazione relativi a tutti o ad alcuni degli oggetti in un sistema, indipendentemente dal fatto che gli oggetti siano stati recentemente manipolati, ad esempio, quando gli eventi di configurazione vengono abilitati per la prima volta.

Considerare l'utilizzo di diversi comandi, ciascuno con una diversa selezione di oggetti, ma in modo che tutti siano inclusi.

- Se si è verificato un errore nel SISTEMA SYSTEM.ADMIN.CONFIG.EVENT . In questa circostanza, non viene generato alcun messaggio di evento di configurazione per gli eventi Crea, Modifica o Elimina. Quando l'errore sulla coda è stato corretto, è possibile utilizzare il comando Gestore code di aggiornamento per richiedere la creazione di messaggi di evento, che sono stati persi mentre si era verificato un errore nella coda. In questa situazione, considerare di impostare l'intervallo di aggiornamento sul tempo per cui la coda non era disponibile.

Concetti correlati

[“Eventi di configurazione” a pagina 42](#)

Gli eventi di configurazione sono notifiche generate quando un oggetto viene creato, modificato o eliminato e possono essere generate anche da richieste esplicite.

Riferimenti correlati

[AGGIORNA QMGR](#)

[Aggiornamento gestore code](#)

Eventi di comandi

Gli eventi di comando sono notifiche che un comando MQSC o PCF è stato eseguito correttamente.

I dati evento contengono le seguenti informazioni:

Informazioni sull'origine

comprende il gestore code da cui è stato emesso il comando, l'ID dell'utente che ha immesso il comando e il modo in cui è stato immesso il comando, ad esempio da un comando della console.

Informazioni contesto

una replica delle informazioni di contesto nei dati del messaggio dal messaggio di comandi. Se un comando non viene immesso utilizzando un messaggio, le informazioni di contesto vengono omesse.

Le informazioni di contesto sono incluse nei dati evento solo quando il comando è stato immesso come messaggio sul SISTEMA SYSTEM.COMMAND.INPUT .

Informazioni sul comando

il tipo di comando immesso.

Dati di comandi

- per i comandi PCF, una replica dei dati del comando
- per comandi MQSC, il testo del comando

Il formato dei dati del comando non corrisponde necessariamente al formato del comando originale. Ad esempio, su [Multiplatforme](#) il formato dei dati del comando è sempre in formato PCF, anche se la richiesta originale era un comando MQSC.

Ogni messaggio di evento di comando generato viene collocato nella coda eventi di comando, SYSTEM.ADMIN.COMMAND.EVENT.

Riferimenti correlati

[Comando](#)

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

Generazione evento comando

Utilizzare questa pagina per visualizzare le situazioni che causano la generazione di eventi di comando e per comprendere le circostanze in cui gli eventi di comando non vengono generati

Un messaggio di evento di comando viene generato nelle situazioni seguenti:

- Quando l'attributo del gestore code CMDEV viene specificato come ENABLED e un comando MQSC o PCF viene eseguito correttamente.
- Quando l'attributo del gestore code CMDEV viene specificato come NODISPLAY e qualsiasi comando viene eseguito correttamente, ad eccezione dei comandi DISPLAY (MQSC) e PCF (Inquire commands).
- Quando si esegue il comando MQSC, ALTER QMGR o il comando PCF, il gestore code di modifica e l'attributo del gestore code CMDEV soddisfano una delle seguenti condizioni:
 - CMDEV non è specificato come DISABLED dopo la modifica
 - CMDEV non è stato specificato come DISABLED prima della modifica

Se un comando viene eseguito sulla coda di eventi comando, SYSTEM.ADMIN.COMMAND.EVENT, viene generato un evento di comando se la coda esiste ancora e non è inibita dall'inserimento.

Quando gli eventi di comando non vengono generati

Un messaggio di evento di comando non viene generato nelle seguenti circostanze:

- Quando un comando ha esito negativo
- Quando un gestore code rileva un errore nel tentativo di inserire un evento comando nella coda eventi, nel qual caso il comando viene eseguito indipendentemente, ma non viene generato alcun messaggio evento
- Per il comando MQSC REFRESH QMGR TYPE (EARLY)
- Per il comando MQSC START QMGR MQSC
- Per il comando MQSC SUSPEND QMGR, se viene specificato il parametro LOG
- Per il comando MQSC RESUME QMGR, se viene specificato il parametro LOG

Concetti correlati

[“Eventi di comandi” a pagina 46](#)

Gli eventi di comando sono notifiche che un comando MQSC o PCF è stato eseguito correttamente.

Riferimenti correlati

[AGGIORNA QMGR](#)

[INIZIO GESTORE CODE](#)

[Gestore code in stato SUSPEND](#)

[RESUME QMGR](#)

Utilizzo evento comando

Utilizzare questa pagina per visualizzare il modo in cui è possibile utilizzare gli eventi di comando per generare una traccia di controllo dei comandi eseguiti

Ad esempio, se un oggetto viene modificato in modo imprevisto, è possibile memorizzare le informazioni relative a chi ha apportato la modifica e quando è stata eseguita. Ciò può essere particolarmente utile quando sono abilitati anche gli eventi di configurazione. Se un comando MQSC o PCF causa la generazione di un evento di comando e di un evento di configurazione, entrambi i messaggi di evento condivideranno lo stesso identificativo di correlazione nel relativo descrittore del messaggio.

Se viene generato un messaggio di evento di comando, ma non può essere inserito nella coda di eventi di comando, ad esempio se la coda di eventi di comando non è stata definita, il comando per cui è stato generato l'evento di comando viene ancora eseguito indipendentemente.

Effetti di CMDSCOPE

Per i comandi in cui viene utilizzato CMDSCOPE, il messaggio o i messaggi di evento del comando verranno generati sul gestore code o sui gestori code in cui viene eseguito il comando, non dove viene immesso il comando. Tuttavia, tutte le informazioni di origine e di contesto nei dati evento saranno correlate al comando originale immesso, anche se il comando che utilizza CMDSCOPE è stato generato dal gestore code di origine.

Concetti correlati

[“Eventi di comandi” a pagina 46](#)

Gli eventi di comando sono notifiche che un comando MQSC o PCF è stato eseguito correttamente.

[“Generazione evento comando” a pagina 47](#)

Utilizzare questa pagina per visualizzare le situazioni che causano la generazione di eventi di comando e per comprendere le circostanze in cui gli eventi di comando non vengono generati

Riferimenti correlati

[Comandi MQSC](#)

[Comandi PCF e risposte in gruppi](#)

Multi **Eventi di registrazione**

Gli eventi del programma di registrazione sono notifiche che un gestore code ha avviato la scrittura in una nuova estensione di log **IBM i**, su IBM i, in un ricevitore di giornale. **z/OS** I messaggi di eventi del programma di registrazione non sono disponibili con IBM MQ for z/OS

I dati evento contengono le seguenti informazioni:

- Il nome dell'estensione log corrente.
- Il nome della prima estensione di log necessaria per il ripristino del riavvio.
- Il nome della prima estensione di log necessaria per il ripristino del supporto.
- La directory in cui si trovano le estensioni di log.
- Il nome della prima estensione di log per cui è richiesta la notifica di archiviazione.

Un evento del programma di registrazione viene generato per la gestione del log di archiviazione, ossia quando **ARCHLOG** cambia e il valore **ARCHLOG** viene incluso nel messaggio dell'evento del programma di registrazione.

Consultare [DISPLAY QMSTATUS](#) per ulteriori informazioni su tutti questi parametri.

È possibile utilizzare i valori **CURRLOG** e **ARCHLOG** per determinare cosa archiviare. Ogni volta che **CURRLOG** cambia, i file con numeri inferiori a **CURRLOG** possono essere inviati per l'archiviazione e, una volta completata l'archiviazione per l'estensione, è necessario richiamare [SET LOG](#) per notificare il gestore code.

ARCHLOG è l'estensione meno recente che richiede l'archiviazione. Una volta archiviata tale estensione e richiamato **SET LOG** per notificare al gestore code che è stato archiviato, il gestore code sposta **ARCHLOG** nella successiva estensione in sequenza. Il gestore code pianifica le estensioni più vecchie di **ARCHLOG**, da eliminare o riutilizzare.

Una volta che **SET LOG** è stato richiamato per questa estensione, viene emesso un nuovo evento, come **ARCHLOG** sarà cambiato, ed è necessario sapere cosa deve essere archiviato dopo il nuovo evento.

Se, per qualche motivo, il processo di archiviazione ha esito negativo e si verifica una grande quantità di notifiche, l'amministratore può immettere manualmente il comando **RESET QMGR TYPE (ARCHLOG)**. Ciò notifica al gestore code che è possibile riutilizzare o eliminare tutte le estensioni precedenti e incluse l'estensione specificata.

Ogni messaggio di evento del programma di registrazione generato viene inserito nella coda eventi del programma di registrazione, SYSTEM.ADMIN.LOGGER.EVENT.

Riferimenti correlati

[Programma di registrazione](#)

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

Generazione evento programma di registrazione

Utilizzare questa pagina per visualizzare le situazioni che causano la generazione di eventi del programma di registrazione e per comprendere le circostanze in cui gli eventi del programma di registrazione non vengono generati

Un messaggio di evento del programma di registrazione viene generato nelle situazioni seguenti:

- Quando l'attributo del gestore code LOGGEREV è specificato come ENABLED e il gestore code inizia a scrivere in una nuova estensione di log o, su IBM i, in un ricevitore di giornale.
- Quando l'attributo del gestore code LOGGEREV viene specificato come ENABLED e il gestore code viene avviato.
- Quando l'attributo del gestore code LOGGEREV viene modificato da DISABLED a ENABLED.
- Quando l'attributo del gestore code LOGGEREV viene notificato che il nome dell'estensione di log meno recente, per cui il gestore code è in attesa di notifica di archivio (ARCHLOG) è stato archiviato.

Suggerimento: È possibile utilizzare il comando RESET QMGR MQSC per richiedere a un gestore code di avviare la scrittura in una nuova estensione log.

Quando gli eventi del programma di registrazione non vengono generati

Un messaggio di evento del programma di registrazione non viene generato nelle seguenti circostanze:



- Quando un gestore code è configurato per utilizzare la registrazione circolare.

In questo caso, l'attributo del gestore code LOGGEREV è impostato come DISABLED e non può essere modificato.

- Quando un gestore code rileva un errore nel tentativo di inserire un evento del programma di registrazione nella coda eventi, in tal caso l'azione che ha causato l'evento viene completata, ma non viene generato alcun messaggio evento.

Concetti correlati

[“Eventi di registrazione” a pagina 48](#)

Gli eventi del programma di registrazione sono notifiche che un gestore code ha avviato la scrittura in una nuova estensione di log  o, su IBM i, in un ricevitore di giornale.  I messaggi di eventi del programma di registrazione non sono disponibili con IBM MQ for z/OS

Riferimenti correlati

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(numero intero con segno a 10 cifre\)](#)

[RESET QMGR](#)

Multi *Utilizzo evento logger*

Utilizzare questa pagina per visualizzare come utilizzare gli eventi del programma di registrazione per determinare le estensioni di log che non sono più richieste per il riavvio del gestore code o per il ripristino del supporto.

È possibile archiviare le estensioni di log superflue su un supporto come il nastro per il ripristino di emergenza prima di rimuoverle dalla directory di log attiva. La rimozione regolare delle estensioni di log superflue mantiene al minimo l'utilizzo dello spazio su disco.

Se l'attributo del gestore code LOGGEREV è abilitato, ma un messaggio di evento del programma di registrazione non può essere inserito nella coda eventi del programma di registrazione, ad esempio perché la coda eventi non è stata definita, l'azione che ha causato l'evento continua a prescindere.

Concetti correlati

[“Eventi di registrazione” a pagina 48](#)

Gli eventi del programma di registrazione sono notifiche che un gestore code ha avviato la scrittura in una nuova estensione di log **IBM i**, su IBM i, in un ricevitore di giornale. **z/OS** I messaggi di eventi del programma di registrazione non sono disponibili con IBM MQ for z/OS

Riferimenti correlati

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(numero intero con segno a 10 cifre\)](#)

[“Generazione evento programma di registrazione” a pagina 49](#)

Utilizzare questa pagina per visualizzare le situazioni che causano la generazione di eventi del programma di registrazione e per comprendere le circostanze in cui gli eventi del programma di registrazione non vengono generati

Multi *Programma C di esempio per monitorare la coda eventi del programma di registrazione*

Utilizzare questa pagina per visualizzare un programma C di esempio che monitora la coda eventi del logger per nuovi messaggi di eventi, legge tali messaggi e inserisce il contenuto del messaggio in 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 <cmqfc.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');
}
}

```

Output di esempio

Questa applicazione produce la seguente forma di output:

```

/*****
/* 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
Log Path                 QMCSIM
-----

```

Concetti correlati

[“Utilizzo evento logger” a pagina 50](#)

Utilizzare questa pagina per visualizzare come utilizzare gli eventi del programma di registrazione per determinare le estensioni di log che non sono più richieste per il riavvio del gestore code o per il ripristino del supporto.

[“Utilizzo evento comando” a pagina 48](#)

Utilizzare questa pagina per visualizzare il modo in cui è possibile utilizzare gli eventi di comando per generare una traccia di controllo dei comandi eseguiti

Riferimenti correlati

[“Generazione evento programma di registrazione” a pagina 49](#)

Utilizzare questa pagina per visualizzare le situazioni che causano la generazione di eventi del programma di registrazione e per comprendere le circostanze in cui gli eventi del programma di registrazione non vengono generati

Eventi di configurazione dell'autorità

Gli eventi di configurazione dell'autorizzazione vengono emessi quando viene effettuata una modifica da una delle operazioni di controllo della sicurezza tramite la riga comandi, MQSC, PCF o i comandi IBM i corrispondenti.

I dati evento contengono le seguenti informazioni:

Informazioni sull'origine

comprende il gestore code da cui è stata apportata la modifica, l'ID dell'utente che ha apportato la modifica e il modo in cui è stata apportata la modifica, ad esempio mediante un comando della console.

Informazioni contesto

una replica delle informazioni di contesto nei dati del messaggio dal messaggio di comandi.

Le informazioni di contesto sono incluse nei dati evento quando il comando è stato immesso come messaggio sul SISTEMA SYSTEM.ADMIN.COMMAND.QUEUE .

Identità record di autorizzazione

comprende il nome profilo e il tipo di oggetto del record di autorizzazione.

Attributi oggetto

comprende i valori di tutti gli attributi nel record di autorizzazione.

Nel caso di eventi record di autorizzazione modifica, vengono generati due messaggi, uno con le informazioni prima della modifica, l'altro con le informazioni dopo la modifica.

Ogni messaggio di evento generato viene inserito nel SISTEMA SYSTEM.ADMIN.CONFIG.EVENT .

Riferimenti correlati

[“Tipi di evento” a pagina 12](#)

Utilizzare questa pagina per visualizzare i tipi di eventi di strumentazione che un gestore code o un'istanza del canale possono segnalare

Generazione evento configurazione autorizzazione

Utilizzare questa pagina per visualizzare le situazioni che causano la generazione di eventi di configurazione dell'autorizzazione e per comprendere le circostanze in cui gli eventi di configurazione dell'autorizzazione non vengono generati.

Gli eventi di configurazione dell'autorizzazione notificano le modifiche agli attributi di un record di autorizzazione. Esistono tre tipi di evento di configurazione dell'autorizzazione:

- [Modifica record di autorizzazione](#)
- [Eliminare il record di autorizzazione](#)
- [Aggiorna record autorizzazione](#)

Un messaggio di evento di autorizzazione viene inserito nella coda di eventi di configurazione, quando l'attributo del gestore code **CONFIGEV** è impostato su *ENABLED* e viene emesso uno dei seguenti

comandi, o il loro equivalente MQSC, anche se non vi è alcuna modifica effettiva al record di autorizzazioni:

- comando PCF [Cancellazione record di autorizzazione](#)
- comando [Imposta record di autorizzazione PCF](#)
- comando di controllo [setmqaut](#)
- Comando CL [RVKMQMAUT](#)
- Comando CL [GRMQMAUT](#)

Quando gli eventi di configurazione dell'autorità non vengono generati

I messaggi di evento di configurazione dell'autorizzazione non vengono generati nelle seguenti circostanze:

- Quando un comando ha esito negativo
- Quando un gestore code rileva un errore nel tentativo di inserire un messaggio nella coda eventi, nel cui caso il comando viene completato, ma non viene generato alcun messaggio evento
- Quando si crea o si elimina un gestore code
- Quando un oggetto viene eliminato indipendentemente dall'opzione **AUTHREC** sul comando di eliminazione. L'evento di comando corrispondente mostra tale operazione, che non si applica al record di autorizzazione per i singoli utenti.

Concetti correlati

“Eventi di comandi” a pagina 46

Gli eventi di comando sono notifiche che un comando MQSC o PCF è stato eseguito correttamente.

Riferimenti correlati

[AGGIORNA QMGR](#)

Programma di esempio per monitorare gli eventi di strumentazione su Multiplatforms

amqsevt formatta gli eventi di strumentazione che un gestore code può creare e viene fornito con IBM MQ for Multiplatforms. Il programma legge i messaggi dalle code eventi e li formatta in stringhe leggibili.

Come programma di esempio, vengono forniti sia l'origine che il binario. L'esempio viene fornito su tutte le piattaforme multiple, incluso IBM i.

Il singolo file binario `amqsevt` (o `amqsevt.exe`) viene fornito nel fileset degli esempi e viene installato nella directory `bin` degli esempi (`tools\c\samples\bin` o `bin64`).

I file di origine `amqsevt.c` vengono forniti anche nel fileset degli esempi e vengono installati nella directory degli esempi, ovvero `tools\c\samples` su Windows.

Notare che il programma può leggere da più code di eventi e sottoscrivere più argomenti, utilizzando [MQCB](#) per recuperare i messaggi.

Quando è in esecuzione come un client, l'esempio può connettersi a qualsiasi gestore code incluso z/OS.

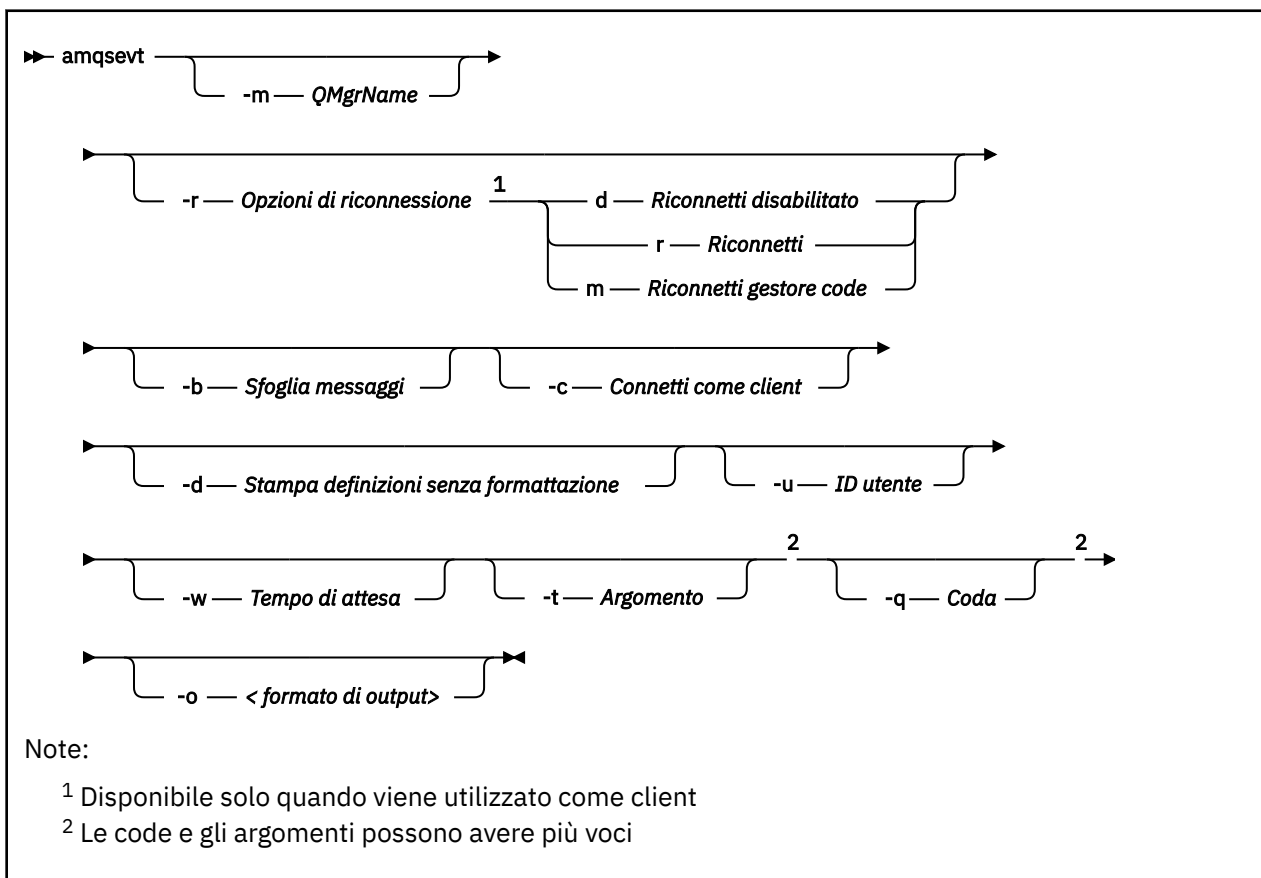


Attenzione: È possibile utilizzare il programma senza specificare alcun parametro, nel qual caso il programma tenta di connettersi al gestore code predefinito e leggere i messaggi dalla serie standard di code eventi (`SYSTEM.ADMIN.*EVENTO`).

In questa situazione, il programma attende per sempre i messaggi, finché non si preme Invio per terminare il programma.

Tuttavia, è più probabile utilizzare il programma con le varie opzioni descritte.

Sintassi



Parametri facoltativi

-m QueueManagerName

Specificare un gestore code specifico per la lettura degli eventi.

-r Reconnection Options

Opzioni di riconnessione automatica quando utilizzate come client. I valori possibili sono:

d

Riconnetti il client disabilitato

r

Riconnetti il client

m

Riconnetti il gestore code

-b

Sfoglia solo i record, piuttosto che leggere in modo distruttivo i messaggi

-c

Seleziona la connessione come client.

-d

Seleziona la modalità di stampa utilizzata nel secondo esempio. Le costanti MQI vengono stampate esattamente come appaiono nei file di intestazione.

-u User ID

Specificare un utente specifico e fare in modo che venga visualizzato un prompt che richiede una password

-w Wait

Fa uscire il programma se nessun messaggio di evento è arrivato entro il numero di secondi specificato.

Si noti che, se non si specifica un'ora, il programma termina normalmente solo quando si preme il tasto Invio.

-t Topic e

-q Queue

Entrambe le opzioni **-q** e **-t** possono essere fornite più volte sulla riga comandi.

Pertanto, è possibile leggere da alcune code standard e anche da argomenti (se gli eventi vengono inviati loro) da una singola esecuzione del programma.

Se sulla riga comandi non viene specificato alcun argomento o coda, vengono aperte le code eventi predefinite.

Nota: Il programma rileva se si è connesso a un gestore code z/OS come client e modifica in modo appropriato la serie predefinita di code eventi, poiché z/OS non dispone del SISTEMA SYSTEM.ADMIN.LOGGER.EVENT .

Quando gli argomenti vengono utilizzati, il programma utilizza una sottoscrizione non durevole con una coda gestita in modo che tutto venga ripulito quando esce.

-o <output format>

Il formato dell'output. I possibili valori sono:

testo

Formato testo standard; questo è il valore predefinito.

json

Formato JSON standard; qualsiasi applicazione che comprende JSON può prendere questo output ed elaborarlo direttamente.

Output di esempio

I seguenti tre esempi mostrano l'output del programma.

Il primo esempio utilizza l'opzione di formattazione predefinita in cui il programma prende la definizione MQI di un campo e formatta l'output per rendere l'output più leggibile.

```
**** 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
```

Il secondo esempio mostra la formattazione alternativa, utilizzando l'opzione **-d**, che non tenta di convertire le costanti MQI. Ciò potrebbe essere preferibile per alcuni strumenti di script che cercano valori MQI specifici.

```
**** 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
```

Il terzo esempio mostra l'output JSON del comando

```
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,
    ...
  }
}

```

Utilizzo di esempio

Il seguente esempio mostra come utilizzare più di una coda:

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

Concetti correlati

[“Monitoraggio eventi” a pagina 9](#)

Il controllo eventi è il processo di rilevamento delle ricorrenze di *eventi di strumentazione* in una rete di gestore code. Un evento di strumentazione è una combinazione logica di eventi rilevati da un gestore code o da un'istanza del canale. Tale evento fa sì che il gestore code o l'istanza del canale inserisci un messaggio speciale, denominato *messaggio evento*, su una coda eventi.

[“Eventi di strumentazione” a pagina 10](#)

Un evento di strumentazione è una combinazione logica di condizioni che un gestore code o un'istanza del canale rileva e inserisce in una coda eventi un messaggio speciale, denominato *messaggio evento*.

Riferimenti correlati

[Programmazione C](#)

[“Programma C di esempio per monitorare la coda eventi del programma di registrazione” a pagina 50](#)

Utilizzare questa pagina per visualizzare un programma C di esempio che monitora la coda eventi del logger per nuovi messaggi di eventi, legge tali messaggi e inserisce il contenuto del messaggio in stdout.

Monitoraggio di messaggi

Il monitoraggio dei messaggi è il processo di identificazione dell'instradamento seguito da un messaggio attraverso una rete di gestori code. Identificando i tipi di attività e la sequenza di attività eseguite per conto di un messaggio, è possibile determinare l'instradamento del messaggio.

Quando un messaggio passa attraverso una rete di gestori code, diversi processi eseguono attività per conto del messaggio. Utilizzare una delle seguenti tecniche per stabilire un instradamento del messaggio:

- L'applicazione di instradamento di visualizzazione IBM MQ (dspmqrte)
- Registrazione attività
- Messaggistica di traccia - instradamento

Queste tecniche generano tutti messaggi speciali che contengono informazioni sulle attività eseguite sul messaggio durante il passaggio attraverso una rete del gestore code. Utilizzare le informazioni restituite in questi messaggi speciali per raggiungere i seguenti obiettivi:

- Registrare l'attività del messaggio.

- Determinare l'ultima ubicazione nota di un messaggio.
- Rileva problemi di instradamento nella rete del gestore code.
- Consente di determinare le cause dei problemi di instradamento nella rete del gestore code.
- Confermare che la rete del gestore code sia in esecuzione correttamente.
- Familiarizzare con l'esecuzione della rete del gestore code.
- Tracciare i messaggi pubblicati.

Concetti correlati

[Tipi di messaggio](#)

Attività e operazioni

Le attività sono azioni discrete che un'applicazione esegue per conto di un messaggio. Le attività sono costituite da operazioni, che sono singole parti di lavoro eseguite da una applicazione.

Le seguenti azioni sono esempi di attività:

- Un MCA (message channel agent) invia un messaggio da una coda di trasmissione in un canale
- Un MCA riceve un messaggio da un canale e lo inserisce nella coda di destinazione
- Un'applicazione che riceve un messaggio da un coda e che invia un messaggio di risposta in risposta.
- Il motore di pubblicazione / sottoscrizione IBM MQ elabora un messaggio.

Le attività sono costituite da una o più *operazioni*. Le operazioni sono singole parti di lavoro eseguite da un'applicazione. Ad esempio, l'attività di un MCA che invia un messaggio da una coda di trasmissione in un canale consiste nelle seguenti operazioni:

1. Richiamo di un messaggio da una coda di trasmissione (un'operazione *Get*).
2. Invio del messaggio su un canale (operazione *Invia*).

In una rete di pubblicazione / sottoscrizione, l'attività del motore di pubblicazione / sottoscrizione IBM MQ che elabora un messaggio può essere costituita dalle seguenti operazioni multiple:

1. Inserimento di un messaggio in una stringa di argomenti (operazione *Put*).
2. Zero o più operazioni per ognuno dei sottoscrittori considerati per la ricezione del messaggio (un'operazione di *pubblicazione*, un'operazione di *pubblicazione scartata* o un'operazione di *pubblicazione esclusa*).

Informazioni dalle attività

È possibile identificare la sequenza di attività eseguite su un messaggio registrando le informazioni quando il messaggio viene instradato attraverso una rete di gestori code. È possibile stabilire l'instradamento di un messaggio attraverso la rete del gestore code dalla sequenza di attività eseguite sul messaggio e ottenere le seguenti informazioni:

L'ultima ubicazione nota di un messaggio

Se un messaggio non raggiunge la destinazione desiderata, è possibile determinare l'ultima ubicazione nota del messaggio da un instradamento del messaggio completo o parziale.

Problemi di configurazione con una rete di gestore code

Quando si studia l'instradamento di un messaggio attraverso una rete di gestori code, è possibile che il messaggio non sia andato dove previsto. Esistono molti motivi per cui ciò può verificarsi, ad esempio, se un canale è inattivo, il messaggio potrebbe prendere un instradamento alternativo.

Per un'applicazione di pubblicazione / sottoscrizione, è anche possibile determinare l'instradamento di un messaggio da pubblicare in un argomento e tutti i messaggi che fluiscono in una rete di gestori code come risultato della pubblicazione ai sottoscrittori.

In tali situazioni, un amministratore di sistema può determinare se vi sono problemi nella rete del gestore code e, se appropriato, correggerli.

Instradamenti messaggi

A seconda del proprio motivo per determinare un instradamento del messaggio, è possibile utilizzare i seguenti approcci generali:

Utilizzo delle informazioni sull'attività registrate per un messaggio di trace - route

I messaggi di instradamento traccia registrano le informazioni sull'attività per uno scopo specifico. È possibile utilizzarli per determinare problemi di configurazione con una rete del gestore code o per determinare l'ultima ubicazione nota di un messaggio. Se viene generato un messaggio di trace - route per determinare l'ultima posizione nota di un messaggio che non ha raggiunto la destinazione desiderata, può imitare il messaggio originale. Ciò fornisce al messaggio di trace - route la maggiore possibilità di seguire l'instradamento preso dal messaggio originale.

L'applicazione di instradamento di visualizzazione IBM MQ può generare messaggi di instradamento traccia.

Utilizzo delle informazioni sull'attività registrate per il messaggio originale

È possibile abilitare qualsiasi messaggio per la registrazione dell'attività e fare in modo che le informazioni sull'attività siano registrate per suo conto. Se un messaggio non raggiunge la destinazione desiderata, è possibile utilizzare le informazioni sull'attività registrata per determinare l'ultima ubicazione nota del messaggio. Utilizzando le informazioni sull'attività dal messaggio originale, è possibile determinare l'instradamento del messaggio più accurato possibile, portando all'ultima posizione nota. Per utilizzare questo approccio, è necessario abilitare il messaggio originale per la registrazione dell'attività.

Avviso: Evitare di abilitare tutti i messaggi in una rete del gestore code per la registrazione dell'attività. I messaggi abilitati per la registrazione attività possono avere molti report di attività generati per loro conto. Se ogni messaggio in una rete di gestori code è abilitato per la registrazione dell'attività, il traffico di rete del gestore code può aumentare ad un livello non accettabile.

Concetti correlati

[“Monitoraggio di messaggi” a pagina 59](#)

Il monitoraggio dei messaggi è il processo di identificazione dell'instradamento seguito da un messaggio attraverso una rete di gestori code. Identificando i tipi di attività e la sequenza di attività eseguite per conto di un messaggio, è possibile determinare l'instradamento del messaggio.

[“Tecniche di instradamento dei messaggi” a pagina 61](#)

La registrazione dell'attività e la messaggistica di traceroute sono tecniche che consentono di registrare le informazioni sull'attività per un messaggio mentre viene instradato attraverso una rete di gestori code.

[“Messaggistica di traccia - instradamento” a pagina 68](#)

La messaggistica di traccia - instradamento è una tecnica che utilizza i *messaggi di traccia - instradamento* per registrare le informazioni sull'attività per un messaggio. La messaggistica di traccia - instradamento implica l'invio di un messaggio di traccia - instradamento in una rete di gestori code.

Attività correlate

[Scrittura dei propri agent del canale dei messaggi](#)

Tecniche di instradamento dei messaggi

La registrazione dell'attività e la messaggistica di traceroute sono tecniche che consentono di registrare le informazioni sull'attività per un messaggio mentre viene instradato attraverso una rete di gestori code.

Registrazione attività

Se per un messaggio è specificata l'opzione di report appropriata, richiede che le applicazioni generino *report di attività* quando vengono instradati attraverso una rete di gestori code. Quando un'applicazione esegue un'attività per conto di un messaggio, è possibile generare un report di attività e consegnarlo a un'ubicazione appropriata. Un report di attività contiene informazioni sull'attività eseguita sul messaggio.

Le informazioni sull'attività raccolte utilizzando i report di attività devono essere organizzate in ordine prima di poter determinare un instradamento del messaggio.

Messaggistica di traccia - instradamento

La *messaggistica di indirizzamento traccia* è una tecnica che implica l'invio di un *messaggio di instradamento traccia* in una rete di gestori code. Quando un'applicazione esegue un'attività per conto del messaggio di traceroute, le informazioni sull'attività possono essere accumulate nei dati del messaggio di traceroute o possono essere generati report di attività. Se le informazioni sull'attività vengono accumulate nei dati del messaggio di traceroute, quando raggiungono la coda di destinazione, è possibile generare un messaggio di risposta di traceroute contenente tutte le informazioni del messaggio di traceroute e consegnarlo a un'ubicazione appropriata.

Poiché un messaggio di trace - route è dedicato alla registrazione della sequenza di attività eseguite per suo conto, ci sono più opzioni di elaborazione disponibili rispetto ai messaggi normali che richiedono i report di attività.

Confronto della registrazione dell'attività e della messaggistica di traccia - instradamento

Sia la registrazione dell'attività che la messaggistica traceroute possono fornire informazioni sull'attività per determinare l'instradamento che un messaggio ha intrapreso attraverso una rete di gestori code. Entrambi i metodi hanno i loro vantaggi.

Vantaggi	Registrazione attività	Messaggistica di traccia - instradamento
È possibile determinare l'ultima posizione nota di un messaggio	Sì	Sì
È possibile determinare i problemi di configurazione con una rete di gestori code	Sì	Sì
Può essere richiesto da qualsiasi messaggio (non è limitato all'utilizzo con i messaggi di traceroute)	Sì	No
I dati del messaggio non vengono modificati	Sì	No
Messaggio elaborato normalmente	Sì	No
Le informazioni sull'attività possono essere accumulate nei dati del messaggio	No	Sì
Consegna messaggio facoltativa alla coda di destinazione	No	Sì
Se un messaggio viene catturato in un loop infinito, può essere rilevato e gestito	No	Sì
Le informazioni sull'attività possono essere messe in ordine in modo affidabile	No	Sì
Applicazione fornita per visualizzare le informazioni sull'attività	No	Sì

Completezza dell'instradamento del messaggio

In alcuni casi non è possibile identificare la sequenza completa di attività eseguite per conto di un messaggio, quindi è possibile determinare solo un instradamento di messaggi parziale. La completezza di un instradamento di messaggi è influenzata direttamente dalla rete del gestore code attraverso cui vengono instradati i messaggi.

Gli MCA e le applicazioni scritte dall'utente connesse ai gestori code possono registrare le informazioni relative alle attività eseguite per conto di un messaggio. La registrazione delle informazioni sulle attività è controllata dagli attributi del gestore code ACTIVREC e ROUTEREC. Una rete di gestori code può determinare instradamenti di messaggi completi.

Modalità di memorizzazione delle informazioni sull'attività

IBM MQ memorizza le informazioni sulle attività nei report di attività, nei messaggi di instradamento traccia o nei messaggi di risposta di instradamento traccia. In ogni caso, le informazioni vengono memorizzate in una struttura denominata gruppo PCF *Attività*. Un messaggio di instradamento traccia o un messaggio di risposta di instradamento traccia può contenere molti gruppi PCF attività, a seconda del numero di attività eseguite sul messaggio. I report di attività contengono un gruppo PCF attività perché viene generato un report di attività separato per ogni attività registrata.

Con la messaggistica trace - route, è possibile registrare ulteriori informazioni. Queste informazioni aggiuntive vengono memorizzate in una struttura denominata gruppo PCF *TraceRoute*. Il gruppo PCF *TraceRoute* contiene un numero di strutture PCF utilizzate per memorizzare ulteriori informazioni sull'attività e per specificare le opzioni che determinano il modo in cui il messaggio di tracerouto viene gestito mentre viene instradato attraverso una rete di gestori code.

Concetti correlati

[“Registrazione attività” a pagina 63](#)

La registrazione dell'attività è una tecnica per determinare gli instradamenti che i messaggi assumono attraverso una rete di gestori code. Per determinare l'instradamento seguito da un messaggio, vengono registrate le attività eseguite per conto del messaggio.

[“Messaggistica di traccia - instradamento” a pagina 68](#)

La messaggistica di traccia - instradamento è una tecnica che utilizza i *messaggi di traccia - instradamento* per registrare le informazioni sull'attività per un messaggio. La messaggistica di traccia - instradamento implica l'invio di un messaggio di traccia - instradamento in una rete di gestori code.

Riferimenti correlati

[“gruppo PCF TraceRoute” a pagina 74](#)

Gli attributi nel gruppo PCF *TraceRoute* controllano il funzionamento di un messaggio di instradamento traccia. Il gruppo PCF *TraceRoute* si trova nei dati del messaggio di ogni messaggio di instradamento traccia.

[“Dati del messaggio del report attività” a pagina 110](#)

Utilizzare questa pagina per visualizzare i parametri contenuti dal gruppo PCF *Attività* in un messaggio del prospetto attività. Alcuni parametri vengono restituiti solo quando sono state eseguite operazioni specifiche.

Registrazione attività

La registrazione dell'attività è una tecnica per determinare gli instradamenti che i messaggi assumono attraverso una rete di gestori code. Per determinare l'instradamento seguito da un messaggio, vengono registrate le attività eseguite per conto del messaggio.

Quando si usa la registrazione di attività, ogni attività eseguita per conto di un messaggio può essere registrata in un report di attività. Un report di attività è un tipo di messaggio di report. Ogni report di attività contiene informazioni sull'applicazione che ha eseguito l'attività per conto del messaggio, quando ha avuto luogo l'attività e informazioni sulle operazioni eseguite come parte dell'attività. I report di attività vengono generalmente consegnati a una coda di risposta in cui vengono raccolti insieme. Studiando i report di attività relativi a un messaggio, è possibile determinare l'instradamento utilizzato dal messaggio attraverso la rete del gestore code.

Utilizzo report attività

Quando i messaggi vengono instradati attraverso una rete di gestori code, è possibile generare report di attività. È possibile utilizzare le informazioni del report attività nei modi seguenti:

Determinare l'ultima ubicazione nota di un messaggio

Se un messaggio abilitato per la registrazione dell'attività non raggiunge la destinazione desiderata, i report di attività generati per il messaggio quando è stato instradato attraverso una rete di gestori code possono essere studiati per determinare l'ultima ubicazione nota del messaggio.

Determinare i problemi di configurazione con una rete di gestori code

Una serie di messaggi abilitati per la registrazione dell'attività possono essere inviati in una rete del gestore code. Studiando i report di attività relativi a ciascun messaggio, può essere evidente che non hanno preso il percorso previsto. Esistono molti motivi per cui ciò può verificarsi, ad esempio, un canale potrebbe essere stato arrestato, forzando il messaggio a prendere un instradamento alternativo. In queste situazioni, un amministratore del sistema può determinare se si sono verificati problemi nella rete del gestore code e, in caso contrario, correggerli.

Nota: È possibile utilizzare la registrazione dell'attività insieme ai messaggi di traceroute utilizzando l'applicazione di instradamento di visualizzazione IBM MQ .

Formato report attività

I report di attività ... sono messaggi PCF generati dalle applicazioni che hanno eseguito un'attività ... per conto di un messaggio. I report di attività sono messaggi di report IBM MQ standard che contengono un descrittore di messaggi e dati di messaggi, come segue:

Il descrittore del messaggio

- Una struttura MQMD

Dati messaggio

- Un'intestazione PCF incorporata (MQEPH)
- Dati del messaggio del report attività

I dati del messaggio del report attività sono costituiti dal gruppo PCF *Attività* e, se generati per un messaggio di traccia - instradamento, dal gruppo PCF *TraceRoute* .

Riferimenti correlati

[MQMD - Descrittore messaggi](#)

[MQEPH - Intestazione PCF integrata](#)

Controllo registrazione attività

Abilitare la registrazione attività a livello di gestore code. Per abilitare un'intera rete di gestori code, abilitare singolarmente ogni gestore code nella rete per la registrazione delle attività. Se si abilitano più gestori code, vengono generati più report di attività.

Informazioni su questa attività

Per generare i report di attività per un messaggio così come viene instradato attraverso un gestore code: definire il messaggio per richiedere i report di attività; abilitare il gestore code per la registrazione delle attività; e assicurarsi che le applicazioni che eseguono attività sul messaggio siano in grado di creare report di attività.

Se non si desidera che i report di attività vengano generati per un messaggio poiché è instradato attraverso un gestore code, *disabilitare* il gestore code per la registrazione dell'attività.

Procedura

1. Richiedere report di attività per un messaggio
 - a) Nel descrittore del messaggio, specificare MQRO_ACTIVITY nel campo *Report* .
 - b) Nel descrittore del messaggio, specificare il nome di una coda di risposta nel campo *ReplyToQ* .

Avviso: Evitare di abilitare tutti i messaggi in una rete del gestore code per la registrazione dell'attività. I messaggi abilitati per la registrazione attività possono avere molti report di attività generati per loro conto. Se ogni messaggio in una rete di gestori code è abilitato per la registrazione dell'attività, il traffico di rete del gestore code può aumentare ad un livello non accettabile.

2. Abilitare o disabilitare il gestore code per la registrazione dell'attività.

Utilizzare il comando MQSC ALTER QMGR, specificando il parametro ACTIVREC, per cambiare il valore dell'attributo gestore code. Il valore può essere:

MSG

Il gestore code è abilitato per la registrazione attività. Tutti i report di attività creati vengono consegnati alla coda di risposta specificata nel descrittore del messaggio. Questo è il valore predefinito.

CODA

Il gestore code è abilitato per la registrazione attività. Tutti i report di attività generati vengono consegnati alla coda del sistema locale SYSTEM.ADMIN.ACTIVITY.QUEUE. La coda di sistema può essere utilizzata anche per inoltrare i report di attività a una coda comune.

DISABILITATO

Il gestore code è disabilitato per la registrazione attività. Non viene generato alcun report di attività nell'ambito di questo gestore code.

Ad esempio, per abilitare un gestore code per la registrazione dell'attività e specificare che i report di attività generati vengono consegnati alla coda del sistema locale SYSTEM.ADMIN.ACTIVITY.QUEUE, utilizzare il seguente comando MQSC:

```
ALTER QMGR ACTIVREC(QUEUE)
```

Attenzione: Quando si modifica l'attributo del gestore code *ACTIVREC*, un MCA in esecuzione non rileva la modifica fino a quando il canale non viene riavviato.

3. Verificare che l'applicazione utilizzi lo stesso algoritmo utilizzato da MCA per determinare se generare un report di attività per un messaggio:
 - a) Verificare che il messaggio abbia richiesto la creazione di report di attività
 - b) Verificare che il gestore code in cui si trova attualmente il messaggio sia abilitato per la registrazione dell'attività
 - c) Inserire il report di attività sulla coda determinato dall'attributo del gestore code *ACTIVREC*

Impostazione di una coda comune per i report di attività

Per determinare le posizioni dei report di attività correlati a un messaggio specifico quando i report vengono consegnati alla coda del sistema locale, è più efficiente utilizzare una coda comune su un singolo nodo

Prima di iniziare

Impostare il parametro **ACTIVREC** per abilitare il gestore code per la registrazione dell'attività e per specificare che i report di attività generati vengono consegnati alla coda del sistema locale SYSTEM.ADMIN.ACTIVITY.QUEUE.

Informazioni su questa attività

Se un certo numero di gestori code in una rete di gestori code sono impostati per consegnare i report di attività alla coda del sistema locale, è possibile che sia necessario del tempo per determinare le ubicazioni dei report di attività correlati a un messaggio specifico. In alternativa, utilizzare un singolo nodo, che è un gestore code che ospita una coda comune. Tutti i gestori code in una rete di gestori code possono consegnare report di attività a questa coda comune. Il vantaggio dell'utilizzo di una coda comune consiste nel fatto che i gestori code non devono consegnare i report di attività alla coda di risposta specificata in un messaggio e, quando si determinano le posizioni dei report di attività correlati a un messaggio, si interroga solo una coda.

Per impostare una coda comune, effettuare quanto segue:

Procedura

1. Selezionare o definire un gestore code come singolo nodo
2. Sul singolo nodo, selezionare o definire una coda da utilizzare come coda comune

3. Su tutti i gestori code in cui i report di attività devono essere consegnati alla coda comune, ridefinire la coda del sistema locale SYSTEM.ADMIN.ACTIVITY.QUEUE come definizione di coda remota:
 - a) Specificare il nome del singolo nodo come nome del gestore code remoto
 - b) Specificare il nome della coda comune come nome della coda remota

Determinazione delle informazioni di instradamento del messaggio

Per determinare un instradamento del messaggio, ottenere le informazioni dai report di attività raccolti. Determinare se nella coda di risposta è presente un numero sufficiente di report di attività per consentire di determinare le informazioni richieste e disporre i report di attività in ordine.

Informazioni su questa attività

L'ordine in cui i report di attività vengono inseriti nella coda di risposta non è necessariamente correlato all'ordine in cui sono state eseguite le attività. È necessario ordinare i report di attività ... manualmente, a meno che non vengano generati per un messaggio di traceroute, nel qual caso è possibile utilizzare l'applicazione di visualizzazione dell'instradamento IBM MQ per ordinare i report di attività ...

Determinare se nella coda di risposta sono presenti report di attività sufficienti per ottenere le informazioni necessarie:

Procedura

1. Identificare tutti i report di attività correlati nella coda di risposta confrontando gli identificativi dei report di attività e il messaggio originale. Assicurarsi di impostare l'opzione di report del messaggio originale in modo che i report di attività possano essere correlati con il messaggio originale.
2. Ordinare i report di attività identificati dalla coda di risposta.
È possibile utilizzare i seguenti parametri dal report di attività:

OperationType

I tipi di operazioni eseguite potrebbero consentire di determinare il report di attività generato direttamente prima o dopo il report di attività corrente.

Ad esempio, un report di attività indica in dettaglio che un MCA ha inviato un messaggio da una coda di trasmissione in un canale. L'ultima operazione dettagliata nel report di attività ha un *OperationType* di send e i dettagli che il messaggio è stato inviato utilizzando il canale, CH1, al gestore code di destinazione, QM1. Ciò significa che la successiva attività eseguita sul messaggio si verificherà sul gestore code QM1 e che sarà iniziata con un'operazione receive dal canale, CH1. Utilizzando queste informazioni è possibile identificare il successivo report di attività, a condizione che esista e sia stato acquisito.

OperationDate e OperationTime

È possibile determinare l'ordine generale delle attività dalle date e ore delle operazioni in ciascun report di attività.

Avviso: A meno che ogni gestore code nella rete del gestore code non abbia gli orologi di sistema sincronizzati, l'ordinamento per data e ora non garantisce che i report di attività siano nella sequenza corretta. È necessario stabilire l'ordine manualmente.

L'ordine dei report di attività rappresenta l'instradamento, o instradamento parziale, che il messaggio ha utilizzato attraverso la rete del gestore code.

3. Ottenere le informazioni necessarie dalle informazioni di attività nei report di attività ordinati.
Se non si dispone di informazioni sufficienti sul messaggio, è possibile acquisire ulteriori report di attività.

Richiamo di ulteriori report di attività

Per stabilire un instradamento del messaggio, è necessario che siano disponibili informazioni sufficienti dai report di attività raccolti. Se si richiamano i report di attività correlati a un messaggio dalla coda

di risposta che il messaggio ha specificato, ma non si dispone delle informazioni necessarie, ricercare ulteriori report di attività.

Informazioni su questa attività

Per determinare le ubicazioni di ulteriori report di attività, effettuare le seguenti operazioni:

Procedura

1. Per i gestori code nella rete del gestore code che consegnano i report di attività a una coda comune, richiamare i report di attività dalla coda comune con un *CorrelId* che corrisponde al *MsgId* del messaggio originale.
2. Per i gestori code nella rete del gestore code che non consegnano i report di attività a una coda comune, richiamare i report di attività nel modo seguente:
 - a) Esaminare i report di attività esistenti per identificare i gestori code attraverso i quali è stato instradato il messaggio.
 - b) Per questi gestori code, identificare quelli abilitati per la registrazione dell'attività.
 - c) Per questi gestori code, identificare quelli che non hanno restituito i report di attività alla coda di risposta specificata.
 - d) Per ogni gestore code identificato, controllare la coda di sistema SYSTEM.ADMIN.ACTIVITY.QUEUE e richiamare tutti i report di attività che hanno un *CorrelId* che corrisponde al *MsgId* del messaggio originale.
 - e) Se non viene rilevato alcun report di attività sulla coda di sistema, controllare la coda di messaggi non recapitabili del gestore code, se ne esiste una.
Un report di attività può essere consegnato a una coda di messaggi non recapitabili solo se l'opzione di report, MQRO_DEAD_LETTER_Q, è impostata.
3. Organizzare tutti i report di attività acquisiti in ordine.
L'ordine dei report di attività rappresenta quindi l'instradamento, o instradamento parziale, che il messaggio ha preso.
4. Ottenere le informazioni necessarie dalle informazioni di attività nei report di attività ordinati.
In alcune circostanze, le informazioni sull'attività registrate non possono raggiungere la coda di risposta specificata, una coda comune o una coda di sistema.

Circostanze in cui le informazioni sull'attività non vengono acquisite

Per stabilire la sequenza completa delle attività eseguite per conto di un messaggio, è necessario acquisire le informazioni relative a ciascuna attività. Se le informazioni relative a qualsiasi attività non sono state registrate o non sono state acquisite, è possibile determinare solo una sequenza parziale di attività.

Le informazioni sull'attività non vengono registrate nelle seguenti circostanze:

- Il messaggio viene elaborato da un gestore code precedente a IBM WebSphere MQ 6.0.
- Il messaggio viene elaborato da un gestore code che non è abilitato per la registrazione dell'attività.
- L'applicazione che prevedeva di elaborare il messaggio non è in esecuzione.

Le informazioni sull'attività registrata non sono in grado di raggiungere la coda di risposta specificata nelle seguenti circostanze:

- Non esiste alcun canale definito per instradare i report di attività alla coda di risposta.
- Il canale per instradare i report di attività alla coda di risposta non è in esecuzione.
- La definizione della coda remota per instradare i report di attività al gestore code in cui risiede la coda di risposta (l'alias del gestore code), non è definita.
- L'utente che ha generato il messaggio originale non dispone dell'autorità di apertura o di inserimento per l'alias del gestore code.

- L'utente che ha generato il messaggio originale non dispone dell'autorizzazione di apertura o di inserimento per la coda di risposta.
- L'inserimento della coda di risposta è inibito.

Le informazioni sull'attività registrate non sono in grado di raggiungere la coda di sistema o una coda comune nelle seguenti circostanze:

- Se deve essere utilizzata una coda comune e non esiste alcun canale definito per instradare i report di attività alla coda comune.
- Se deve essere utilizzata una coda comune e il canale per instradare i report di attività alla coda comune non è in esecuzione.
- Se deve essere utilizzata una coda comune e la coda di sistema non è definita correttamente.
- L'utente che ha generato il messaggio originale non dispone dell'autorità di apertura o di inserimento per la coda di sistema.
- L'immissione della coda di sistema è inibita.
- Se deve essere utilizzata una coda comune e l'utente che ha generato il messaggio originale non dispone dell'autorità di apertura o di inserimento per la coda comune.
- Se una coda comune deve essere utilizzata e la coda comune non è inserita.

In queste condizioni, se il report di attività non ha l'opzione di report MQRO_DISCARD_MSG specificata, il report di attività può essere richiamato da una coda di messaggi non recapitabili se ne è stata definita una sul gestore code in cui il report di attività è stato rifiutato. Un report di attività avrà questa opzione di report specificata solo se il messaggio originale, da cui è stato creato il report di attività, ha MQRO_PASS_DISCARD_AND_SCADENZA e MQRO_DISCARD_MSG specificati nel campo Report del descrittore del messaggio.

Messaggistica di traccia - instradamento

La messaggistica di traccia - instradamento è una tecnica che utilizza i *messaggi di traccia - instradamento* per registrare le informazioni sull'attività per un messaggio. La messaggistica di traccia - instradamento implica l'invio di un messaggio di traccia - instradamento in una rete di gestori code.

Poiché il messaggio di instradamento della traccia viene instradato attraverso la rete del gestore code, le informazioni sull'attività vengono registrate. Queste informazioni sulle attività includono le informazioni sulle applicazioni che hanno eseguito le attività, quando sono state eseguite e le operazioni che sono state eseguite come parte delle attività. È possibile utilizzare le informazioni registrate utilizzando la messaggistica di traccia - instradamento per i seguenti scopi:

Per determinare l'ultima posizione nota di un messaggio

Se un messaggio non raggiunge la destinazione desiderata, è possibile utilizzare le informazioni sull'attività registrate per un messaggio di traceroute per determinare l'ultima ubicazione nota del messaggio. Un messaggio di traccia - route viene inviato in una rete del gestore code con la stessa destinazione del messaggio originale, con l'intenzione di seguire lo stesso instradamento. Le informazioni sull'attività possono essere accumulate nei dati del messaggio di traccia - instradamento o registrate utilizzando i report di attività. Per aumentare la probabilità che il messaggio di traccia - instradamento segua lo stesso instradamento del messaggio originale, è possibile modificare il messaggio di traccia - instradamento per imitare il messaggio originale.

Per stabilire problemi di configurazione con una rete di gestori code

I messaggi di traccia - instradamento vengono inviati in una rete di gestori code e le informazioni sull'attività vengono registrate. Studiando le informazioni sull'attività registrate per un messaggio di traccia - instradamento, può diventare evidente che il messaggio di traccia - instradamento non ha seguito l'instradamento previsto. Ci sono molti motivi per cui ciò può verificarsi, ad esempio, un canale potrebbe essere inattivo, forzando il messaggio a prendere un instradamento alternativo. In queste situazioni, un amministratore del sistema può determinare se si sono verificati problemi nella rete del gestore code e, in caso contrario, correggerli.

È possibile utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per configurare, generare e inserire messaggi di instradamento traccia in una rete di gestori code.

Avviso: Se si inserisce un messaggio di instradamento traccia in un elenco di distribuzione, i risultati non sono definiti.

Concetti correlati

[“Riferimento messaggio di traccia - instradamento” a pagina 128](#)

Utilizzare questa pagina per ottenere una panoramica del formato del messaggio di traccia - route. I dati del messaggio di traccia - instradamento includono parametri che descrivono le attività che il messaggio di traccia - instradamento ha causato

Modalità di registrazione delle informazioni sull'attività

Con la messaggistica di traccia - instradamento, è possibile registrare le informazioni sull'attività nei dati del messaggio di traccia - instradamento oppure utilizzare i report di attività. In alternativa, è possibile utilizzare entrambe le tecniche.

Accumulazione delle informazioni sull'attività nei dati del messaggio di traccia - instradamento

Quando un messaggio di traceroute viene instradato attraverso una rete di gestore code, le informazioni sulle attività eseguite per conto del messaggio di traceroute possono essere accumulate nei dati del messaggio di traceroute. Le informazioni sull'attività vengono memorizzate nei gruppi PCF *Attività* . Per ogni attività eseguita per conto del messaggio di instradamento traccia, un gruppo PCF *Attività* viene scritto alla fine del blocco PCF nei dati del messaggio del messaggio di instradamento traccia.

Ulteriori informazioni sull'attività vengono registrate nella messaggistica traccia - route, in un gruppo PCF denominato gruppo PCF *TraceRoute* . Le informazioni di attività aggiuntive vengono memorizzate in questo gruppo PCF e possono essere utilizzate per determinare la sequenza delle attività registrate. Questa tecnica è controllata dal parametro *Accumulate* nel gruppo PCF *TraceRoute* .

Registrazione delle informazioni sull'attività utilizzando i report di attività

Quando un messaggio di traceroute viene instradato attraverso una rete di gestori code, è possibile generare un report di attività per ogni attività eseguita per conto del messaggio di traceroute. Le informazioni sull'attività vengono memorizzate nel gruppo PCF *Attività* . Per ogni attività eseguita per conto di un messaggio di traccia - instradamento, viene generato un report di attività contenente un gruppo PCF *Attività* . La registrazione dell'attività per i messaggi di traccia - instradamento funziona allo stesso modo di qualsiasi altro messaggio.

I report di attività generati per i messaggi di traccia - instradamento contengono ulteriori informazioni sull'attività rispetto a quelle generate per qualsiasi altro messaggio. Le informazioni aggiuntive vengono restituite in un gruppo *TraceRoute* PCF. Le informazioni contenute nel gruppo PCF *TraceRoute* sono accurate solo dal momento in cui è stato generato il report di attività. È possibile utilizzare le informazioni aggiuntive per determinare la sequenza delle attività eseguite per conto del messaggio di traccia - instradamento.

Acquisizione delle informazioni sulle attività registrate

Quando un messaggio di traceroute ha raggiunto la destinazione desiderata o viene eliminato, il metodo utilizzato per acquisire le informazioni sull'attività dipende dal modo in cui tali informazioni sono state registrate.

Prima di iniziare

Se non si ha familiarità con le informazioni sulle attività, fare riferimento a [“Modalità di registrazione delle informazioni sull'attività” a pagina 69](#).

Informazioni su questa attività

Utilizzare i seguenti metodi per acquisire le informazioni sull'attività dopo che il messaggio di traceroute ha raggiunto la destinazione prevista o è stato eliminato:

Procedura

- Richiamare il messaggio trace - route.

Il parametro *Trasferisci*, nel gruppo PCF *TraceRoute*, controlla se un messaggio di instradamento della traccia viene posizionato sulla coda di destinazione all'arrivo o se viene eliminato. Se il messaggio di traceroute viene consegnato alla coda di destinazione, è possibile richiamare il messaggio di traceroute da questa coda. Quindi, è possibile utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per visualizzare le informazioni sull'attività.

Per richiedere che le informazioni sull'attività vengano raccolte nei dati del messaggio di un messaggio di traceroute, impostare il parametro *Accumulate* nel gruppo PCF *TraceRoute* su MQRROUTE_ACCUMULATE_IN_MSG.

- Utilizzare un messaggio di risposta traceroute.

Quando un messaggio di traceroute raggiunge la destinazione desiderata o il messaggio di traceroute non può essere instradato ulteriormente in una rete del gestore code, è possibile generare un messaggio di risposta traceroute. Un messaggio di risposta all'instradamento della traccia contiene un duplicato di tutte le informazioni sull'attività dal messaggio di instradamento della traccia e viene consegnato a una coda di risposta specificata o alla coda di sistema SYSTEM.ADMIN.TRACE.ROUTE.QUEUE. È possibile utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per visualizzare le informazioni sull'attività.

Per richiedere un messaggio di risposta traceroute, impostare il parametro *Accumulate* nel gruppo PCF *TraceRoute* su MQRROUTE_ACCUMULATE_AND_REPLY.

- Utilizzare i report di attività.

Se i report di attività vengono generati per un messaggio di traceroute, è necessario individuare i report di attività prima di poter acquisire le informazioni sull'attività. Quindi, per stabilire la sequenza di attività, è necessario ordinare i report di attività.

Controllo della messaggistica di traccia - instradamento

Abilitare la messaggistica di traceroute a livello di gestore code, in modo che le applicazioni nell'ambito di tale gestore code possano scrivere informazioni sull'attività in un messaggio di traceroute. Per abilitare un'intera rete di gestori code, abilitare singolarmente ogni gestore code nella rete per la messaggistica di instradamento traccia. Se si abilitano più gestori code, vengono generati più report di attività.

Prima di iniziare

Se si utilizzano i report di attività per registrare le informazioni di attività per un messaggio di traceroute, fare riferimento a [“Controllo registrazione attività” a pagina 64](#).

Informazioni su questa attività

Per registrare le informazioni sull'attività per un messaggio di trace - route come instradato tramite un gestore code, effettuare le seguenti operazioni:

Procedura

- Definire il modo in cui le informazioni sull'attività devono essere registrate per il messaggio di trace - route.

Fare riferimento a [“Creazione e configurazione di un messaggio di trace - route” a pagina 73](#)

- Se si desidera accumulare le informazioni sull'attività nel messaggio traceroute, assicurarsi che il gestore code sia abilitato per la messaggistica traceroute
- Se si desidera accumulare le informazioni sull'attività nel messaggio di traccia - instradamento, assicurarsi che le applicazioni che eseguono le attività sul messaggio di traccia - instradamento siano in grado di scrivere le informazioni sull'attività nei dati del messaggio di traccia - instradamento

Concetti correlati

[“Creazione e configurazione di un messaggio di trace - route” a pagina 73](#)

Un messaggio di trace - route comprende parti specifiche di dati di messaggi e descrittori di messaggi. Per generare un messaggio di indirizzamento di traccia, creare il messaggio manualmente oppure utilizzare l'applicazione di indirizzamento di visualizzazione IBM MQ .

Attività correlate

“Controllo registrazione attività” a pagina 64

Abilitare la registrazione attività a livello di gestore code. Per abilitare un'intera rete di gestori code, abilitare singolarmente ogni gestore code nella rete per la registrazione delle attività. Se si abilitano più gestori code, vengono generati più report di attività.

Abilitazione dei gestori code per la messaggistica di trace - route

Per controllare se i gestori code sono abilitati o disabilitati per la messaggistica di trace - route, utilizzare l'attributo del gestore code ROUTEREC.

Utilizzare il comando MQSC ALTER QMGR, specificando il valore ROUTEREC per modificare il valore dell'attributo gestore code. Il valore può essere uno dei seguenti:

MSG

Il gestore code è abilitato per la messaggistica di traccia - instradamento. Le applicazioni nell'ambito del gestore code possono scrivere le informazioni sull'attività nel messaggio traceroute.

Se il parametro *Accumulate* nel gruppo PCF *TraceRoute* è impostato come MQROUTE_ACCUMULATE_AND_REPLYe l'attività successiva da eseguire sul messaggio di tracerouti:

- è un'eliminazione
- è un inserimento in una coda locale (coda di destinazione o coda di messaggi non instradabili)
- farà sì che il numero totale di attività eseguite sul messaggio tracerout superi il valore del parametro *MaxActivities*, nel gruppo PCF *TraceRoute* .

viene generato un messaggio di risposta di trace - route e consegnato alla coda reply - to specificata nel descrittore del messaggio di trace - route.

CODA

Il gestore code è abilitato per la messaggistica di traccia - instradamento. Le applicazioni nell'ambito del gestore code possono scrivere le informazioni sull'attività nel messaggio traceroute.

Se il parametro *Accumulate* nel gruppo PCF *TraceRoute* è impostato come MQROUTE_ACCUMULATE_AND_REPLYe l'attività successiva da eseguire sul messaggio di tracerouti:

- è un'eliminazione
- è un inserimento in una coda locale (coda di destinazione o coda di messaggi non instradabili)
- farà sì che il numero totale di attività eseguite sul messaggio tracerout superi il valore del parametro *MaxActivities*, nel gruppo PCF *TraceRoute* .

viene generato un messaggio di risposta di trace - route e consegnato alla coda del sistema locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE.

DISABILITATO

Il gestore code è stato disabilitato per la messaggistica trace - route. Le informazioni sull'attività non vengono accumulate nel messaggio di traceroute, tuttavia il gruppo PCF *TraceRoute* può essere aggiornato nell'ambito di questo gestore code.

Ad esempio, per disabilitare un gestore code per la messaggistica trace - route, utilizzare il seguente comando MQSC:

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

Attenzione: Quando si modifica l'attributo del gestore code *ROUTEREC* , un MCA in esecuzione non rileva la modifica finché il canale non viene riavviato.

Abilitazione delle applicazioni per la messaggistica di traccia - instradamento

Per abilitare la messaggistica di traceroute per un'applicazione utente, basare l'algoritmo sull'algoritmo utilizzato dagli MCA (message channel agent)

Prima di iniziare

Se non si ha familiarità con il formato di un messaggio di traceroute, consultare [“Riferimento messaggio di traccia - instradamento”](#) a pagina 128.

Informazioni su questa attività

Gli agent MCA (Message Channel Agent) sono abilitati per la messaggistica trace - route. Per abilitare un'applicazione utente per la messaggistica di trace - route, utilizzare la seguente procedura dall'algoritmo utilizzato dagli MCA:

Procedura

1. Determinare se il messaggio elaborato è un messaggio di indirizzamento traccia.
Se il messaggio non è conforme al formato di un messaggio di trace - route, il messaggio non viene elaborato come un messaggio di trace - route.
2. Determinare se le informazioni sull'attività devono essere registrate.
Se il livello di dettagli dell'attività eseguita non è inferiore al livello di dettagli specificato dal parametro *Dettagli*, le informazioni sull'attività vengono registrate in circostanze specifiche. Queste informazioni vengono registrate solo se l'accumulo delle richieste di messaggi di traccia - instradamento e il gestore code sono abilitati per la messaggistica di traccia - instradamento o se il messaggio di traccia - instradamento richiede un report di attività e il gestore code è abilitato per la registrazione dell'attività.
 - Se le informazioni sull'attività devono essere registrate, incrementare il parametro *RecordedActivities*.
 - Se le informazioni sull'attività non devono essere registrate, incrementare il parametro *UnrecordedActivities*.
3. Determinare se il numero totale di attività eseguite sul messaggio di instradamento della traccia supera il valore del parametro *MaxActivities*.
Il numero totale di attività è la somma di *RecordedActivities*, *UnrecordedActivities* e *DiscontinuityCount*.
Se il numero totale di attività supera *MaxActivities*, rifiutare il messaggio con feedback MQFB_MAX_ACTIVITIES.
4. Se il valore di *Accumulate* è impostato come MQROUTE_ACCUMULATE_IN_MSG o MQROUTE_ACCUMULATE_AND_REPLY, e il gestore code è abilitato per la messaggistica di trace - route, scrivere un gruppo PCF di attività alla fine del blocco PCF nei dati del messaggio di un messaggio di trace - route.
5. Consegnare il messaggio di trace - route a una coda locale.
 - Se il parametro, *Consegna*, è specificato come MQROUTE_DELIVER_NO, rifiutare il messaggio traceroute con feedback MQFB_NOT_LIVER.
 - Se il parametro, *Deliver*, è specificato come MQROUTE_DELIVER_YES, consegnare il messaggio trace - route alla coda locale.
6. Generare un messaggio di risposta di traceroute se tutte le seguenti condizioni sono vere:
 - Il messaggio di traceroute è stato recapitato ad una coda locale o rifiutato
 - Il valore del parametro, *Accumulate*, è MQROUTE_ACCUMULATE_AND_REPLY
 - Il gestore code è abilitato per la messaggistica di traccia - instradamentoIl messaggio di risposta traceroute viene inserito nella coda determinata dall'attributo del gestore code ROUTEREC.
7. Se il messaggio di tracerout ha richiesto un report di attività e il gestore code è abilitato per la registrazione dell'attività, generare un report di attività.

Il report di attività viene inserito nella coda determinata dall'attributo del gestore code ACTIVREC.

Creazione e configurazione di un messaggio di trace - route

Un messaggio di trace - route comprende parti specifiche di dati di messaggi e descrittori di messaggi. Per generare un messaggio di indirizzamento di traccia, creare il messaggio manualmente oppure utilizzare l'applicazione di indirizzamento di visualizzazione IBM MQ .

Un messaggio di trace - route è composto dalle parti seguenti:

Descrittore messaggio

Una struttura MQMD, con il campo *Formato* impostato su MQFMT_ADMIN o MQFMT_EMBEDDED_PCF.

Dati messaggio

Una delle seguenti combinazioni:

- Un'intestazione PCF (MQCFH) e i dati del messaggio di instradamento della traccia, se *Formato* è impostato su MQFMT_ADMIN
- Un'intestazione PCF integrata (MQEPH), i dati del messaggio di trace - route e ulteriori dati del messaggio specificati dall'utente, se *Formato* è impostato su MQFMT_EMBEDDED_PCF

I dati del messaggio di trace - route sono costituiti dal gruppo PCF *TraceRoute* e da uno o più gruppi PCF *Attività* .

Generazione manuale

Quando si genera un messaggio di tracerouto manualmente, non è richiesto un gruppo PCF *Attività* . I gruppi *Attività* PCF vengono scritti nei dati del messaggio di traccia - instradamento quando un'applicazione MCA o scritta dall'utente esegue un'attività per suo conto.

L'applicazione di instradamento di visualizzazione IBM MQ

Utilizzare l'applicazione di instradamento di visualizzazione IBM MQ , **dspmqrte**, per configurare, generare e inserire un messaggio di instradamento di traccia in una rete di gestori code. Impostare il parametro *Formato* nel descrittore del messaggio su MQFMT_ADMIN. Non è possibile aggiungere dati utente al messaggio di traccia - instradamento generato dall'applicazione di visualizzazione instradamento IBM MQ .

Limitazione: dspmqrte non può essere emesso su gestori code IBM MQ for z/OS . Se si desidera che il primo gestore code attraverso il quale viene instradato il messaggio di traceroute sia un gestore code di questo tipo, connettersi al gestore code come client utilizzando il parametro facoltativo -c.

Imitazione del messaggio originale

Quando si utilizza un messaggio di trace - route per determinare l'instradamento che un altro messaggio ha preso attraverso una rete del gestore code, più un messaggio di trace - route imita il messaggio originale, maggiore è la possibilità che il messaggio di trace - route segua lo stesso instradamento del messaggio originale.

Le seguenti caratteristiche del messaggio possono influire sul punto in cui un messaggio viene inoltrato all'interno di una rete di gestori code:

Priorità

La priorità può essere specificata nel descrittore del messaggio.

Persistenza

La permanenza può essere specificata nel descrittore del messaggio.

Scadenza

La scadenza può essere specificata nel descrittore del messaggio.

Opzioni di documentazione

Le opzioni di prospetto possono essere specificate nel descrittore del messaggio.

Dimensione messaggio

Per imitare la dimensione di un messaggio, è possibile scrivere ulteriori dati nei dati del messaggio. Per questo scopo, ulteriori dati di messaggio possono essere privi di significato.

Suggerimento: L'applicazione di instradamento di visualizzazione IBM MQ non può specificare la dimensione del messaggio.

Dati messaggio

Alcune reti di gestori code utilizzano l'instradamento basato sul contenuto per determinare dove vengono inoltrati i messaggi. In questi casi, i dati del messaggio di traceroute devono essere scritti per imitare i dati del messaggio originale.

Suggerimento: L'applicazione di instradamento di visualizzazione IBM MQ non può specificare i dati del messaggio.

gruppo PCF TraceRoute

Gli attributi nel gruppo PCF *TraceRoute* controllano il funzionamento di un messaggio di instradamento traccia. Il gruppo PCF *TraceRoute* si trova nei dati del messaggio di ogni messaggio di instradamento traccia.

La seguente tabella elenca i parametri nel gruppo *TraceRoute* riconosciuti da un MCA. È possibile aggiungere ulteriori parametri se le applicazioni scritte dall'utente vengono scritte per riconoscerle, come descritto in "Ulteriori informazioni sull'attività" a pagina 79.

Parametro	Tipo
TraceRoute	MQCFGR
Dettaglio	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
Accumula	MQCFIN
Avanti	MQCFIN
Distribuzione	MQCFIN

Le descrizioni di ciascun parametro nel gruppo PCF *TraceRoute* sono le seguenti:

Dettaglio

Specifica il livello di dettaglio delle informazioni sull'attività da registrare. Il valore può essere uno dei seguenti:

MQROUTE_DETAIL_LOW

Vengono registrate solo le attività eseguite dall'applicazione utente.

MQROUTE_DETAIL_MEDIO

Le attività specificate in **MQROUTE_DETAIL_LOW** devono essere registrate. Inoltre, vengono registrate le attività eseguite dagli MCA.

MQROUTE_DETAIL_HIGH

È necessario registrare le attività specificate in **MQROUTE_DETAIL_LOW** e **MQROUTE_DETAIL_MEDIUM**. Gli MCA non registrano ulteriori informazioni sull'attività a questo livello di dettagli. Questa opzione è disponibile solo per le applicazioni utente che devono registrare ulteriori informazioni sull'attività. Ad esempio, se un'applicazione utente determina l'instradamento di un messaggio considerando determinate caratteristiche del messaggio, le informazioni sulla logica di instradamento potrebbero essere incluse con questo livello di dettagli.

RecordedActivities

Specifica il numero di attività registrate eseguite per conto del messaggio di traccia - instradamento. Un'attività viene considerata registrata se le informazioni su di essa sono state scritte nel messaggio di traccia - instradamento o se è stato generato un report di attività. Per ogni attività registrata, *RecordedActivities* aumenta di uno.

UnrecordedActivities

Specifica il numero di attività non registrate eseguite per conto del messaggio di traccia - instradamento. Un'attività viene considerata non registrata se un'applicazione abilitata per la messaggistica di traccia - instradamento non si accumula né scrive le informazioni relative all'attività in un report di attività.

Un'attività eseguita per conto di un messaggio di traccia - instradamento non viene registrata nelle seguenti circostanze:

- Il livello di dettaglio dell'attività ... eseguita è inferiore al livello di dettaglio specificato dal parametro *Dettaglio*.
- Il messaggio di traccia - route richiede un report di attività, ma non l'accumulazione, e il gestore code non è abilitato per la registrazione dell'attività.
- Il messaggio di traccia - route richiede l'accumulo ma non un report di attività e il gestore code non è abilitato per la messaggistica di traccia - route.
- Il messaggio di traccia - route richiede sia un accumulo che un report di attività e il gestore code non è abilitato per la registrazione delle attività e la messaggistica di traccia route.
- Il messaggio di traccia - route non richiede né l'accumulo né un report di attività.

Per ogni attività non registrata, il parametro, *UnrecordedActivities*, aumenta di uno.

DiscontinuityCount

Specifica il numero di volte in cui il messaggio di traceroute è stato instradato attraverso un gestore code con applicazioni non abilitate per la messaggistica traceroute. Questo valore viene incrementato dal gestore code. Se questo valore è maggiore di 0, è possibile determinare solo un instradamento del messaggio parziale.

MaxActivities

Specifica il numero massimo di attività che possono essere eseguite per conto del messaggio di traccia - route.

Il numero totale di attività è la somma di *RecordedActivities*, *UnrecordedActivities* e *DiscontinuityCount*. Il numero totale di attività non deve superare il valore di *MaxActivities*.

Il valore di *MaxActivities* può essere:

Un numero intero positivo

Il numero massimo di attività.

Se il numero massimo di attività viene superato, il messaggio di traccia - route viene rifiutato con il feedback MQFB_MAX_ACTIVITIES. Ciò può impedire che il messaggio di traccia - route venga inoltrato indefinitamente se viene rilevato in un loop infinito.

MQROUTE_UNLIMITED_ACTIVITIES

È possibile eseguire un numero illimitato di attività per conto del messaggio di traccia - instradamento.

Accumula

Specifica il metodo utilizzato per accumulare le informazioni sull'attività. Il valore può essere uno dei seguenti:

MQROUTE_ACCUMULATE_IN_MSG

Se il gestore code è abilitato per la messaggistica di traccia - instradamento, le informazioni sull'attività vengono accumulate nei dati del messaggio di traccia - instradamento.

Se questo valore viene specificato, i dati del messaggio di traccia - route sono i seguenti:

- Gruppo PCF *TraceRoute* .
- Zero o più gruppi PCF *Attività* .

MQROUTE_ACCUMULATE_E_REPLY

Se il gestore code è abilitato per la messaggistica di traccia - instradamento, le informazioni sull'attività vengono accumulate nei dati del messaggio di traccia - instradamento e viene

generato un messaggio di risposta di traccia - instradamento se si verifica una delle seguenti condizioni:

- Il messaggio di traceroute viene eliminato da un gestore code IBM MQ .
- Il messaggio di traceroute viene inserito in una coda locale (coda di destinazione o coda di messaggi non recapitabili) da un gestore code IBM MQ .
- Il numero di attività eseguite sul messaggio di instradamento della traccia supera il valore di *MaxActivities*.

Se questo valore viene specificato, i dati del messaggio di trace - route sono i seguenti:

- Gruppo PCF *TraceRoute* .
- Zero o più gruppi PCF *Attività* .

MQROUTE_ACCUMULATE_NONE

Le informazioni sull'attività non vengono accumulate nei dati del messaggio di traccia - instradamento.

Se questo valore viene specificato, i dati del messaggio di trace - route sono i seguenti:

- Gruppo PCF *TraceRoute* .

Avanti

Specifica dove può essere inoltrato un messaggio di traceroute. Il valore può essere:

MQROUTE_FORWARD_IF_SUPPORTED

Il messaggio trace - route viene inoltrato solo ai gestori code che onoreranno il valore del parametro *Deliver* del gruppo *TraceRoute* .

MQROUTE_FORWARD_ALL

Il messaggio di trace - route viene inoltrato a qualsiasi gestore code, indipendentemente dal fatto che il valore del parametro *Deliver* verrà rispettato.

I gestori code utilizzano il seguente algoritmo quando determinano se inoltrare un messaggio di instradamento della traccia a un gestore code remoto:

1. Determinare se il gestore code remoto è in grado di supportare la messaggistica di instradamento traccia.
 - Se il gestore code remoto è in grado di supportare la messaggistica di instradamento della traccia, l'algoritmo continua con il passo [“4” a pagina 76](#).
 - Se il gestore code remoto non è in grado di supportare la messaggistica di trace - route, l'algoritmo continua con il passo [“2” a pagina 76](#)
2. Determinare se il parametro *Consegna* del gruppo *TraceRoute* contiene eventuali opzioni di consegna non riconosciute nella maschera di bit MQROUTE_DELIVER_REJ_UNSUP_MASK.
 - Se vengono trovate opzioni di consegna non riconosciute, il messaggio trace - route viene rifiutato con il feedback MQFB_UNSUPPORTED_DELIVERY.
 - Se non vengono trovate opzioni di consegna non riconosciute, l'algoritmo continua il passo [“3” a pagina 76](#).
3. Determinare il valore del parametro *Trasferisci* dal gruppo PCF *TraceRoute* nel messaggio tracerout.
 - Se *Consegna* è specificato come MQROUTE_DELIVER_YES, il messaggio di trace - route viene inoltrato al gestore code remoto.
 - Se *Trasferisci* è specificato come MQROUTE_DELIVER_NO, l'algoritmo continua con il passo [“4” a pagina 76](#).
4. Determinare se il parametro *Inoltra* del gruppo *TraceRoute* contiene opzioni di inoltra non riconosciute nella maschera di bit MQROUTE_FORWARDING_REJ_UNSUP_MASK.
 - Se vengono trovate delle opzioni di inoltra non riconosciute, il messaggio traceroute viene rifiutato con il feedback MQFB_UNSUPPORTED_FORWARDING.

- Se non vengono trovate opzioni di inoltra non riconosciute, l'algoritmo continua il passo "5" a pagina 77.
5. Determinare il valore del parametro *Inoltra* dal gruppo PCF *TraceRoute* nel messaggio traceroute.
- Se *Inoltra* è specificato come MQROUTE_FORWARD_IF_SUPPORTED, il messaggio di trace - route viene rifiutato con il feedback MQFB_NOT_FORWARDED.
 - Se *Inoltra* è specificato come MQROUTE_FORWARD_ALL, il messaggio traceroute può essere inoltrato al gestore code remoto.

Distribuzione

Specifica l'azione da intraprendere se il messaggio di trace - route raggiunge la destinazione desiderata. Le applicazioni scritte dall'utente devono controllare questo attributo prima di inserire un messaggio di traceroute nella relativa coda di destinazione. Il valore può essere uno dei seguenti:

MQROUTE_DELIVER_SÌ

All'arrivo, il messaggio di trace - route viene inserito nella coda di destinazione. Qualsiasi applicazione che esegue un'operazione get sulla coda di destinazione può richiamare il messaggio traceroute.

MQROUTE_DELIVER_NO

All'arrivo, il messaggio di trace - route non viene consegnato alla coda di destinazione. Il messaggio viene elaborato in base alle relative opzioni di report.

Impostazione di una coda comune per i messaggi di risposta traceroute

Per determinare le ubicazioni dei messaggi di risposta di traccia - instradamento correlati ad un messaggio specifico quando i report vengono consegnati alla coda del sistema locale, è più efficiente utilizzare una coda comune su un singolo nodo

Prima di iniziare

Impostare il parametro **ROUTEREC** per abilitare il gestore code per la messaggistica di instradamento traccia e per specificare che tutti i messaggi di risposta di instradamento traccia generati vengono consegnati alla coda del sistema locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE.

Informazioni su questa attività

Se un numero di gestori code in una rete di gestori code è impostato per consegnare i messaggi di risposta di traccia - instradamento alla coda del sistema locale, può essere necessario del tempo per determinare le ubicazioni dei messaggi di risposta di traccia - instradamento correlati a un messaggio specifico. In alternativa, utilizzare un singolo nodo, che è un gestore code che ospita una coda comune. Tutti i gestori code in una rete di gestori code possono consegnare messaggi di risposta traceroute a questa coda comune. Il vantaggio dell'utilizzo di una coda comune è che i gestori code non devono consegnare i messaggi di risposta di traccia - instradamento alla coda di risposta specificata in un messaggio e, quando si determinano le ubicazioni dei messaggi di risposta di traccia - instradamento correlati a un messaggio, si interroga solo una coda.

Per impostare una coda comune, effettuare quanto segue:

Procedura

1. Selezionare o definire un gestore code come singolo nodo
2. Sul singolo nodo, selezionare o definire una coda da utilizzare come coda comune
3. Su tutti i gestori code che inoltrano i messaggi di risposta di traccia - instradamento alla coda comune, ridefinire la coda del sistema locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE come definizione di coda remota
 - a) Specificare il nome del singolo nodo come nome del gestore code remoto
 - b) Specificare il nome della coda comune come nome della coda remota

Acquisizione e utilizzo delle informazioni registrate

Utilizzare una delle tecniche riportate di seguito per acquisire le informazioni di attività registrate per un messaggio di indirizzamento traccia

Tenere presente che le circostanze in cui le informazioni sull'attività non vengono acquisite si applicano anche ai messaggi di risposta traceroute.

Le informazioni sull'attività non vengono registrate quando un messaggio di traceroute viene elaborato da un gestore code disabilitato sia per la registrazione dell'attività che per la messaggistica di traceroute.

Acquisizione di informazioni dai messaggi di risposta di traceroute

Per acquisire informazioni sull'attività, individuare il messaggio di risposta traceroute. Quindi, richiamare il messaggio e analizzare le informazioni sull'attività.

Informazioni su questa attività

È possibile acquisire informazioni sull'attività da un messaggio di risposta traceroute solo se si conosce l'ubicazione del messaggio di risposta traceroute. Individuare il messaggio ed elaborare le informazioni sull'attività nel modo seguente:

Procedura

1. Controllare la coda di risposta che è stata specificata nel descrittore del messaggio di instradamento traccia. Se il messaggio di risposta di trace - route non si trova nella coda di reply - to, controllare le seguenti posizioni:
 - La coda del sistema locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, sul gestore code di destinazione del messaggio di trace - route
 - La coda comune, se è stata impostata una coda comune per i messaggi di risposta traceroute
 - La coda del sistema locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, su qualsiasi altro gestore code nella rete del gestore code, che può verificarsi se il messaggio trace - route è stato inserito in una coda di messaggi non instradabili o se è stato superato il numero massimo di attività
2. Richiama il messaggio di risposta di trace - route
3. Utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per visualizzare le informazioni sull'attività registrata
4. Studia le informazioni sull'attivit ... e ottieni le informazioni di cui hai bisogno

Acquisizione delle informazioni dai messaggi di traccia - instradamento

Per acquisire informazioni sull'attività, individuare il messaggio di traccia - instradamento, che deve avere i parametri appropriati nel gruppo PCF *TraceRoute* . Quindi, richiamare il messaggio e analizzare le informazioni sull'attività.

Informazioni su questa attività

È possibile acquisire le informazioni sull'attività da un messaggio di traceroute solo se si conosce l'ubicazione del messaggio di traceroute e dispone del parametro *Accumulate* nel gruppo PCF *TraceRoute* specificato come MQRROUTE_ACCUMULATE_IN_MSG o MQRROUTE_ACCUMULATE_AND_REPLY.

Affinché il messaggio di instradamento della traccia venga consegnato alla coda di destinazione, è necessario specificare il parametro *Deliver* nel gruppo PCF *TraceRoute* come MQRROUTE_DELIVER_YES.

Procedura

1. Controllare la coda di destinazione. Se il messaggio di traceroute non si trova nella coda di destinazione, è possibile tentare di individuare il messaggio di traceroute utilizzando un messaggio di traceroute abilitato per la registrazione dell'attività. Con i report di attività generati, provare a determinare l'ultima ubicazione nota del messaggio traceroute.
2. Richiama il messaggio trace - route

3. Utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per visualizzare le informazioni sull'attività registrata
4. Studia le informazioni sull'attivit ... e ottieni le informazioni di cui hai bisogno

Acquisizione di informazioni dai report di attività

Per acquisire le informazioni sull'attività, individuare il report di attività, che deve avere l'opzione del report specificata nel descrittore del messaggio. Quindi, richiamare il report di attività e analizzare le informazioni sull'attività.

Informazioni su questa attività

È possibile acquisire informazioni sull'attività da un report di attività solo se si conosce l'ubicazione del report di attività e l'opzione di report MQRO_ACTIVITY è stata specificata nel descrittore del messaggio di traccia - instradamento.

Procedura

1. Individuare e ordinare i report di attività generati per un messaggio di traccia - instradamento.
Una volta individuati i report di attività, è possibile ordinarli manualmente oppure utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per ordinare e visualizzare automaticamente le informazioni sull'attività.
2. Studia le informazioni sull'attivit ... e ottieni le informazioni di cui hai bisogno

Ulteriori informazioni sull'attività

Quando un messaggio di traceroute viene instradato attraverso una rete del gestore code, le applicazioni utente possono registrare ulteriori informazioni includendo uno o più parametri PCF aggiuntivi durante la scrittura del gruppo *Attività* nei dati del messaggio di traceroute o del report di attività.

Ulteriori informazioni sull'attività possono aiutare gli amministratori di sistema ad identificare l'instradamento preso da un messaggio di traccia - instradamento o il motivo per cui tale instradamento è stato preso.

Se si utilizza l'applicazione di visualizzazione dell'instradamento IBM MQ per visualizzare le informazioni registrate per un messaggio di traccia - instradamento, qualsiasi parametro PCF aggiuntivo può essere visualizzato solo con un identificativo numerico, a meno che l'identificativo del parametro di ciascun parametro non sia riconosciuto dall'applicazione di visualizzazione dell'instradamento IBM MQ . Per riconoscere un identificativo di parametro, è necessario registrare ulteriori informazioni utilizzando i seguenti parametri PCF. Includere questi parametri PCF in una posizione appropriata nel gruppo PCF *Attività* .

GroupName

Tabella 13. Nome gruppo	
Descrizione	Parametri raggruppati che specificano le informazioni aggiuntive.
Identificativo	MQGACF_VALUE_NAMING.
Tipo dati	MQCFGR
Parametri nel gruppo	<i>ParameterName</i> <i>ParameterValue</i>

ParameterName

Tabella 14. Nome del parametro	
Descrizione	Contiene il nome che deve essere visualizzato dall'applicazione IBM MQ display route, che inserisce il valore di <i>ParameterValue</i> nel contesto.
Identificativo	MQCA_VALUE_NOME.

<i>Tabella 14. Nome del parametro (Continua)</i>	
Descrizione	Contiene il nome che deve essere visualizzato dall'applicazione IBM MQ display route, che inserisce il valore di <i>ParameterValue</i> nel contesto.
Tipo dati	MQCFST
Incluso nel gruppo PCF:	<i>GroupName</i> .
Valore:	Il nome da visualizzare.

ParameterValue

<i>Tabella 15. Valore parametro</i>	
Descrizione	Contiene il valore che deve essere visualizzato dall'applicazione di instradamento di visualizzazione IBM MQ .
Identificativo:	L'identificativo della struttura PCF per le informazioni aggiuntive.
Tipo di dati:	Il tipo di dati della struttura PCF per le informazioni aggiuntive.
Incluso nel gruppo PCF:	<i>GroupName</i> .
Valore:	Il valore da visualizzare.

Esempi di registrazione di ulteriori informazioni sull'attività

I seguenti esempi illustrano come un'applicazione utente può registrare ulteriori informazioni quando si esegue un'attività per conto di un messaggio di traceroute. In entrambi gli esempi, l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzata per creare un messaggio di instradamento traccia e visualizzare le informazioni sull'attività restituite.

Registrazione di ulteriori informazioni sull'attività: Esempio 1

Ulteriori informazioni sull'attività vengono registrate da un'applicazione utente in un formato in cui l'identificativo del parametro *non* è riconosciuto dall'applicazione di instradamento di visualizzazione IBM MQ .

1. L'applicazione di instradamento di visualizzazione IBM MQ è utilizzata per generare e inserire un messaggio di instradamento di traccia in una rete del gestore code. Le opzioni necessarie sono impostate per richiedere quanto segue:
 - Le informazioni sull'attività vengono accumulate nei dati del messaggio di traccia - instradamento.
 - All'arrivo nella coda di destinazione, il messaggio di traccia - instradamento viene eliminato e un messaggio di risposta di traccia - instradamento viene generato e consegnato a una coda di risposta specificata.
 - Al ricevimento del messaggio di risposta di tracerouto, l'applicazione di visualizzazione dell'instradamento IBM MQ visualizza le informazioni sull'attività accumulata.

Il messaggio di trace - route viene inserito nella rete del gestore code.

2. Poiché il messaggio di trace - route viene instradato attraverso la rete del gestore code, un'applicazione utente, abilitata per la messaggistica di trace - route, esegue un'attività di dettaglio bassa per conto del messaggio. Oltre a scrivere le informazioni sull'attività standard nel messaggio di traccia - instradamento, l'applicazione utente scrive il parametro PCF seguente alla fine del gruppo Attività:

ColorValue

Identificativo
65536

Tipo dati

MQCFST

Valore

'Rosso'

Questo parametro PCF aggiuntivo fornisce ulteriori informazioni sull'attività eseguita, tuttavia viene scritto in un formato in cui l'identificativo del parametro *non* è riconosciuto dall'applicazione di visualizzazione instradamento IBM MQ .

3. I messaggi di instradamento traccia raggiungono la coda di destinazione e un messaggio di risposta di instradamento traccia viene restituito all'applicazione di instradamento di visualizzazione IBM MQ . Le informazioni aggiuntive sull'attività vengono visualizzate come segue:

```
65536: 'Red'
```

L'applicazione di instradamento video IBM MQ non riconosce l'identificativo del parametro PCF e lo visualizza come valore numerico. Il contesto delle informazioni aggiuntive non è chiaro.

Per un esempio di quando l'applicazione di visualizzazione instradamento IBM MQ riconosce l'identificativo del parametro PCF, consultare [“Registrazione di ulteriori informazioni sull'attività: Esempio 2”](#) a pagina 81.

Registrazione di ulteriori informazioni sull'attività: Esempio 2

Ulteriori informazioni sull'attività vengono registrate da un'applicazione utente in un formato in cui l'identificativo del parametro viene riconosciuto dall'applicazione di instradamento di visualizzazione IBM MQ .

1. L'applicazione di instradamento di visualizzazione IBM MQ è utilizzata per creare e inserire un messaggio di instradamento della traccia in una rete di gestore code nello stesso modo di [“Registrazione di ulteriori informazioni sull'attività: Esempio 1”](#) a pagina 80.
2. Poiché il messaggio di trace - route viene instradato attraverso la rete del gestore code, un'applicazione utente, abilitata per la messaggistica di trace - route, esegue un'attività di dettaglio bassa per conto del messaggio. Oltre a scrivere le informazioni sull'attività standard nel messaggio di traccia - instradamento, l'applicazione utente scrive i seguenti parametri PCF alla fine del gruppo di attività:

ColorInfo

<i>Tabella 16. Informazioni sul colore</i>	
Descrizione	Parametri raggruppati che specificano le informazioni su un colore.
Identificativo:	MQGACF_VALUE_NAMING.
Tipo di dati:	MQCFGR.
Parametri nel gruppo:	<i>ColorName</i> <i>ColorValue</i>

ColorName

<i>Tabella 17. Nome colore</i>	
Descrizione	Contiene il nome che deve visualizzare l'applicazione di instradamento di visualizzazione IBM MQ che inserisce il valore di <i>ColorValue</i> nel contesto.
Identificativo:	MQCA_VALUE_NOME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>ColorInfo.</i>

Tabella 17. Nome colore (Continua)	
Descrizione	Contiene il nome che deve visualizzare l'applicazione di instradamento di visualizzazione IBM MQ che inserisce il valore di <i>ColorValue</i> nel contesto.
Valore:	'Colore'

ColorValue

Tabella 18. Valore colore	
Descrizione	Contiene il valore che deve essere visualizzato dall'applicazione di instradamento di visualizzazione IBM MQ .
Identificativo:	65536.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>ColorInfo</i> .
Valore:	'Rosso'


Questi parametri PCF aggiuntivi forniscono ulteriori informazioni sull'attività eseguita. Questi parametri PCF vengono scritti in un formato in cui l'identificativo del parametro viene riconosciuto dall'applicazione di visualizzazione instradamento IBM MQ .

3. I messaggi di instradamento traccia raggiungono la coda di destinazione e un messaggio di risposta di instradamento traccia viene restituito all'applicazione di instradamento di visualizzazione IBM MQ . Le informazioni aggiuntive sull'attività vengono visualizzate come segue:

```
Color: 'Red'
```

L'applicazione di instradamento di visualizzazione IBM MQ riconosce che l'identificativo del parametro della struttura PCF contenente il valore delle informazioni aggiuntive sull'attività ha un nome corrispondente. Viene visualizzato il nome corrispondente invece del valore numerico.

IBM MQ visualizza applicazione di instradamento

Utilizzare l'applicazione di instradamento di visualizzazione IBM MQ (**dspmqrte**) per gestire i messaggi di trace - route e le informazioni di attività relative a un messaggio di trace - route, utilizzando una CLI (command - line interface).  L'applicazione di instradamento di visualizzazione IBM MQ non viene fornita su IBM MQ for z/OS, ma è possibile eseguirla da un'installazione distribuita e connetterla come client a un gestore code IBM MQ for z/OS specificando il parametro **-c** quando si immette il comando **dspmqrte** .

È possibile utilizzare l'applicazione di instradamento di visualizzazione IBM MQ **dspmqrte** per i seguenti scopi:

- Per configurare, creare e inserire un messaggio di trace - route in una rete di gestori code.

Inserendo un messaggio di traceroute in una rete di gestori code, è possibile raccogliere e utilizzare le informazioni sull'attività per determinare l'instradamento utilizzato dal messaggio di traceroute. È possibile specificare le caratteristiche dei messaggi di traceroute come segue:

- La destinazione del messaggio di trace - route.
- Il modo in cui il messaggio trace - route imita un altro messaggio.
- Come deve essere gestito il messaggio di trace - route quando viene instradato attraverso una rete di gestori code.
- Indica se la registrazione dell'attività o la messaggistica di traccia - instradamento vengono utilizzate per registrare le informazioni sull'attività.

- Per ordinare e visualizzare le informazioni sull'attività correlate a un messaggio di traccia - instradamento.

Se l'applicazione di instradamento di visualizzazione IBM MQ ha inserito un messaggio di instradamento della traccia in una rete del gestore code, dopo che sono state restituite le informazioni sull'attività correlate, le informazioni possono essere ordinate e visualizzate immediatamente. In alternativa, l'applicazione di visualizzazione dell'instradamento IBM MQ può essere utilizzata per ordinare e visualizzare le informazioni sull'attività relative a un messaggio di traccia - instradamento precedentemente creato.

Riferimenti correlati

[dspmqrte](#)

Parametri per i messaggi di traccia - instradamento

Utilizzare questa pagina per ottenere una panoramica dei parametri forniti dall' IBM MQ applicazione di visualizzazione instradamento **dspmqrte**, per stabilire le caratteristiche di un messaggio di traccia - instradamento, incluso il modo in cui viene considerato come instradato attraverso una rete di gestori code.

Riferimenti correlati

[dspmqrte](#)

Connessione gestore code

Utilizzare questa pagina per specificare il gestore code a cui si connette l'applicazione di instradamento di visualizzazione IBM MQ

-c

Specifica che l'applicazione di instradamento di visualizzazione IBM MQ si connette come applicazione client.

Se non si specifica questo parametro, l'applicazione di instradamento di visualizzazione IBM MQ non si connette come applicazione client.

-m QMgrName

Il nome del gestore code a cui si connette l'applicazione di instradamento di visualizzazione IBM MQ . Il nome può contenere un massimo di 48 caratteri.

Se non si specifica questo parametro, viene utilizzato il gestore code predefinito.

La destinazione di destinazione

Utilizzare questa pagina per specificare la destinazione di un messaggio di traccia - route

-q TargetQName

Se l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzata per inviare un messaggio di instradamento della traccia in una rete del gestore code, *TargetQName* specifica il nome della coda di destinazione.

-ts TargetTopicStringa

Specifica la stringa argomento.

-qm TargetQMGr

Qualifica la destinazione di destinazione; verrà quindi applicata la normale risoluzione del nome del gestore code. La destinazione è specificata con *-q TargetQName* o *-ts TargetTopicString* .

Se non si specifica questo parametro, il gestore code a cui è connessa l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzato come gestore code di destinazione.

-o

Specifica che la destinazione non è collegata ad una destinazione specifica. Generalmente, questo parametro viene utilizzato quando il messaggio di tracersouto deve essere inserito in un cluster. La destinazione di destinazione viene aperta con opzione MQOO_BIND_NOT_FIXED.

Se non si specifica questo parametro, la destinazione di destinazione è collegata a una destinazione specifica.

L'argomento della pubblicazione

Per le applicazioni di pubblicazione / sottoscrizione, utilizzare questa pagina per specificare la stringa di argomenti di un messaggio di traceroute per l'applicazione di instradamento di visualizzazione IBM MQ da pubblicare

-ts TopicName

Specifica una stringa di argomenti in cui l'applicazione di instradamento di visualizzazione IBM MQ deve pubblicare un messaggio di instradamento traccia e inserisce questa applicazione in modalità argomento. In questa modalità, l'applicazione traccia tutti i messaggi che risultano dalla richiesta di pubblicazione.

È inoltre possibile utilizzare l'applicazione di visualizzazione instradamento IBM MQ per visualizzare i risultati di un report di attività generato per la pubblicazione dei messaggi.

Imitazione del messaggio

Utilizzare questa pagina per configurare un messaggio traceroute per imitare un messaggio, ad esempio quando il messaggio originale non ha raggiunto la destinazione prevista

Un utilizzo della messaggistica trace - route è quello di aiutare a determinare l'ultima posizione nota di un messaggio che non ha raggiunto la destinazione prevista. L'applicazione di visualizzazione instradamento IBM MQ fornisce parametri che consentono di configurare un messaggio di instradamento traccia per imitare il messaggio originale. Quando si imita un messaggio, è possibile utilizzare i parametri seguenti:

-l Persistenza

Specifica la persistenza del messaggio trace - route generato. I valori possibili per *Persistenza* sono:

si

Il messaggio di trace - route generato è persistente. (MQPER_PERSISTENT).

no

Il messaggio di traccia - instradamento generato **non** è persistente. (MQPER_NOT_PERSISTENT).

q

Il messaggio traceroute generato eredita il suo valore di persistenza dalla destinazione specificata da *-q TargetQName* o *-ts TargetTopicString*. (MQPER_PERSISTENCE_AS_Q_DEF).

Un messaggio di risposta di traceroute o qualsiasi messaggio di report restituito condividerà lo stesso valore di persistenza del messaggio di traceroute originale.

Se *Persistenza* viene specificato come **si**, è necessario specificare il parametro *-rq ReplyToQ*. La coda di risposta non deve essere risolta in una coda dinamica temporanea.

Se non si specifica questo parametro, il messaggio traceroute generato **non** è persistente.

-p Priorità

Specifica la priorità del messaggio di traccia - instradamento. Il valore di *Priorità* è maggiore o uguale a 0 oppure MQPRI_PRIORITY_AS_Q_DEF. MQPRI_PRIORITY_AS_Q_DEF specifica che il valore della priorità viene preso dalla destinazione specificata da *-q TargetQName* o *-ts TargetTopicString*.

Se non si specifica questo parametro, il valore della priorità viene preso dalla destinazione specificata da *-q TargetQName* o *-ts TargetTopicString*.

-xs Scadenza

Specifica il tempo di scadenza per il messaggio di instradamento traccia, in secondi.

Se non si specifica questo parametro, il tempo di scadenza viene specificato come 60 secondi.

-ro none |ReportOption

none

Specifica che non è impostata alcuna opzione di prospetto.

ReportOption

Specifica le opzioni di prospetto per il messaggio di instradamento traccia. È possibile specificare più opzioni di report utilizzando una virgola come separatore. I valori possibili per *ReportOption* sono:

attività

L'opzione di report MQRO_ACTIVITY è impostata.

Coa

L'opzione di report MQRO_COA_WITH_FULL_DATA è impostata.

COD

L'opzione di report MQRO_COD_WITH_FULL_DATA è impostata.

eccezione

L'opzione di report MQRO_EXCEPTION_WITH_FULL_DATA è impostata.

scadenza

L'opzione di report MQRO_EXPIRATION_WITH_FULL_DATA è impostata.

annullare

L'opzione di report MQRO_DISCARD_MSG è impostata.

Se non viene specificato né *-ro ReportOption* né *-ro none* , vengono specificate le opzioni di report MQRO_ACTIVITY e MQRO_DISCARD_MSG.

L'applicazione di instradamento di visualizzazione IBM MQ non consente di aggiungere dati utente al messaggio di instradamento traccia. Se si desidera che i dati utente vengano aggiunti al messaggio di traccia - instradamento, è necessario generare manualmente il messaggio di traccia - instradamento.

Informazioni sulle attività registrate

Utilizzare questa pagina per specificare il metodo utilizzato per restituire le informazioni sull'attività registrata, che è possibile utilizzare per stabilire l'instradamento utilizzato da un messaggio di traccia - instradamento

Le informazioni di attività registrate possono essere restituite nel modo seguente:

- Nei report di attività
- In un messaggio di risposta di trace - route
- Nel messaggio di trace - route (essendo stato inserito nella coda di destinazione)

Quando si utilizza **dspmqrte**, il metodo utilizzato per restituire le informazioni sull'attività registrata viene determinato utilizzando i seguenti parametri:

-ro attività

Specifica che le informazioni sull'attività vengono restituite utilizzando i report di attività. Per default, la registrazione dell'attività è abilitata.

-ac -ar

Specifica che le informazioni sull'attività vengono accumulate nel messaggio di instradamento traccia e che deve essere generato un messaggio di risposta di instradamento traccia.

-ac

Specifica che le informazioni sull'attività devono essere accumulate all'interno del messaggio trace - route.

Se non si specifica questo parametro, le informazioni sull'attività non vengono accumulate all'interno del messaggio di traceroute.

-ar

Richiede che un messaggio di risposta di traceroute contenente tutte le informazioni di attività accumulate venga generato nelle seguenti circostanze:

- Il messaggio traceroute viene eliminato da un gestore code IBM MQ .
- Il messaggio di traceroute viene inserito in una coda locale (coda di destinazione o coda di messaggi non recapitabili) da un gestore code IBM MQ .
- Il numero di attività eseguite sul messaggio di trace - route supera il valore specificato in *-s Activities*.

-ac -d sì

Specifica che le informazioni sull'attività vengono accumulate nel messaggio di traccia - instradamento e che all'arrivo, il messaggio di traccia - instradamento verrà inserito nella coda di destinazione.

-ac

Specifica che le informazioni sull'attività devono essere accumulate all'interno del messaggio trace - route.

Se non si specifica questo parametro, le informazioni sull'attività non vengono accumulate all'interno del messaggio di traceroute.

-d sì

All'arrivo, il messaggio di trace - route viene inserito nella coda di destinazione, anche se il gestore code non supporta la messaggistica di trace - route.

Se non si specifica questo parametro, il messaggio di traceroute non viene inserito nella coda di destinazione.

Il messaggio di traceroute può quindi essere richiamato dalla coda di destinazione e le informazioni sull'attività registrata possono essere acquisite.

È possibile combinare questi metodi come richiesto.

Inoltre, il livello di dettaglio delle informazioni di attività registrate può essere specificato utilizzando il seguente parametro:

-t *Dettagli*

Specifica le attività registrate. I valori possibili per *Dettaglio* sono:

basso

Le attività eseguite dall'applicazione definita dall'utente vengono registrate solo.

supporto

Le attività specificate in basso vengono registrate. Inoltre, vengono registrate le attività di pubblicazione e le attività eseguite dagli MCA.

alto

Le attività specificate in basso e medio vengono registrate. Gli MCA non espongono ulteriori informazioni sull'attività a questo livello di dettaglio. Questa opzione è disponibile solo per le applicazioni definite dall'utente che devono esporre ulteriori informazioni sull'attività. Ad esempio, se un'applicazione definita dall'utente determina l'instradamento di un messaggio considerando determinate caratteristiche del messaggio, la logica di instradamento potrebbe essere inclusa con questo livello di dettaglio.

Se non si specifica questo parametro, vengono registrate le attività di livello medio.

Per impostazione predefinita, l'applicazione di instradamento di visualizzazione IBM MQ usa una coda dinamica temporanea per memorizzare i messaggi restituiti. Quando l'applicazione di instradamento di visualizzazione IBM MQ termina, la coda dinamica temporanea viene chiusa e tutti i messaggi vengono eliminati. Se i messaggi restituiti sono richiesti oltre l'esecuzione corrente dell'applicazione di instradamento di visualizzazione IBM MQ termina, è necessario specificare una coda permanente utilizzando i seguenti parametri:

-rq *ReplyToQ*

Specifica il nome della coda di risposta a cui vengono inviate tutte le risposte al messaggio di instradamento traccia. Se il messaggio di trace - route è persistente, o se viene specificato il parametro *-n*, è necessario specificare una coda di risposta che non sia una coda dinamica temporanea.

Se non si specifica questo parametro, viene creata una coda di risposta dinamica utilizzando la coda modello predefinita del sistema, SYSTEM.DEFAULT.MODEL.QUEUE.

-rqm *ReplyTo gestore code*

Specifica il nome del gestore code in cui risiede la coda di risposta. Il nome può contenere un massimo di 48 caratteri.

Se non si specifica questo parametro, il gestore code a cui è connessa l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzato come gestore code di risposta.

Modalità di gestione del messaggio di trace - route

Utilizzare questa pagina per controllare come viene gestito un messaggio di trace - route mentre viene instradato attraverso una rete di gestori code.

I seguenti parametri possono limitare la posizione in cui il messaggio di traceroute può essere instradato nella rete del gestore code:

-d Consegna

Specifica se il messaggio di trace - route deve essere consegnato alla coda di destinazione all'arrivo. I valori possibili per *Consegna* sono:

sì

All'arrivo, il messaggio di trace - route viene inserito nella coda di destinazione, anche se il gestore code non supporta la messaggistica di trace - route.

no

All'arrivo, il messaggio di trace - route non viene inserito nella coda di destinazione.

Se non si specifica questo parametro, il messaggio di traceroute non viene inserito nella coda di destinazione.

-f Avanti

Specifica il tipo di gestore code a cui può essere inoltrato il messaggio di trace - route. Per i dettagli sull'algoritmo utilizzato dai gestori code per determinare se inoltrare un messaggio a un gestore code remoto, fare riferimento a [“gruppo PCF TraceRoute” a pagina 74](#). I valori possibili per *Avanti* sono:

tutti

Il messaggio di trace - route viene inoltrato a qualsiasi gestore code.

Avviso: Se inoltrato a un gestore code precedente a IBM WebSphere MQ 6.0, il messaggio di traceroute non verrà riconosciuto e potrà essere consegnato a una coda locale nonostante il valore del parametro *-d Deliver*.

supportati

Il messaggio di trace - route viene inoltrato solo a un gestore code che onorerà il parametro *Deliver* dal gruppo *PCF TraceRoute*.

Se non si specifica questo parametro, il messaggio di traceroute verrà inoltrato solo a un gestore code che rispetterà il parametro *Trasferisci*.

I seguenti parametri possono impedire che un messaggio di traceroute rimanga indefinitamente in una rete del gestore code:

-s Attività

Specifica il numero massimo di attività registrate che possono essere eseguite per conto del messaggio di traccia - instradamento prima che venga eliminato. Ciò impedisce che il messaggio di trace - route venga inoltrato indefinitamente se viene rilevato in un loop infinito. Il valore di *Attività* è maggiore o uguale a 1 o *MQROUTE_UNLIMITED_ACTIVITIES*. *MQROUTE_UNLIMITED_ACTIVITIES* specifica che è possibile eseguire un numero illimitato di attività per conto del messaggio di traceroute.

Se non si specifica questo parametro, è possibile eseguire un numero illimitato di attività per conto del messaggio di traceroute.

-xs Scadenza

Specifica il tempo di scadenza per il messaggio di instradamento traccia, in secondi.

Se non si specifica questo parametro, il tempo di scadenza viene specificato come 60 secondi.

-xp PassExpiry

Specifica se il tempo di scadenza dal messaggio di instradamento traccia viene passato a un messaggio di risposta di instradamento traccia. I valori possibili per *PassExpiry* sono:

si

L'opzione di report MQRO_PASS_DISCARD_AND_SCADENZA è specificata nel descrittore del messaggio di tracerouti.

Se viene generato un messaggio di risposta di trace - route o report di attività per il messaggio di trace - route, l'opzione di report MQRO_DISCARD (se specificata) e il tempo di scadenza rimanente vengono passati.

Questo è il valore predefinito.

no

L'opzione di report MQRO_PASS_DISCARD_AND_SCADENZA non è specificata.

Se viene generato un messaggio di risposta di trace - route per il messaggio di trace - route, l'opzione di eliminazione e il tempo di scadenza dal messaggio di trace - route **non** vengono trasmessi.

Se non si specifica questo parametro, MQRO_PASS_DISCARD_AND_SCADENZA non è specificato.

-ro elimina

Specifica l'opzione di report MQRO_DISCARD_MSG. Ciò può impedire che il messaggio di traceroute rimanga indefinitamente nella rete del gestore code.

Visualizzazione delle informazioni sull'attività

L'applicazione di instradamento di visualizzazione IBM MQ può visualizzare le informazioni sull'attività per un messaggio di instradamento della traccia che ha appena inserito in una rete del gestore code oppure può visualizzare le informazioni sull'attività per un messaggio di instradamento della traccia generato in precedenza. Può anche visualizzare ulteriori informazioni registrate dalle applicazioni scritte dall'utente.

Per specificare se vengono visualizzate le informazioni sull'attività restituite per un messaggio di trace - route, specificare il seguente parametro:

-n

Specifica che le informazioni sull'attività restituite per il messaggio di traccia - instradamento non devono essere visualizzate.

Se questo parametro è accompagnato da una richiesta per un messaggio di risposta traceroute (*-ar*) o da una qualsiasi delle opzioni di generazione del report da (*-ro ReportOption*), è necessario specificare una coda di risposta specifica (non modello) utilizzando *-rq ReplyToQ* . Per impostazione predefinita, sono richiesti solo i messaggi di report di attività.

Una volta inserito il messaggio di instradamento della traccia nella coda di destinazione specificata, viene visualizzata una stringa esadecimale di 48 caratteri contenente l'identificativo del messaggio di instradamento della traccia. L'identificativo del messaggio può essere utilizzato dall'applicazione di visualizzazione dell'instradamento IBM MQ per visualizzare le informazioni sull'attività per il messaggio di traccia - instradamento in un secondo momento, utilizzando il parametro *-i CorrelId* .

Se non si specifica questo parametro, le informazioni sull'attività restituite per il messaggio di traceroute vengono visualizzate nel modulo specificato dal parametro *-v* .

Quando si visualizzano le informazioni sull'attività per un messaggio di traceroute appena inserito in una rete del gestore code, è possibile specificare il seguente parametro:

-w WaitTime

Specifica il tempo, in secondi, durante il quale l'applicazione di visualizzazione dell'instradamento IBM MQ attenderà i report di attività o un messaggio di risposta dell'instradamento della traccia, per tornare alla coda di risposta specificata.

Se non si specifica questo parametro, il tempo di attesa viene specificato come tempo di scadenza del messaggio di trace - route, più 60 secondi.

Quando si visualizzano le informazioni sull'attività accumulate in precedenza, è necessario impostare i seguenti parametri:

-q TargetQName

Se l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzata per visualizzare le informazioni sull'attività raccolte in precedenza, *TargetQName* specifica il nome della coda in cui sono memorizzate le informazioni sull'attività.

-i CorrelId

Questo parametro viene utilizzato quando l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzata solo per visualizzare le informazioni sull'attività accumulate precedentemente. Nella coda specificata da *-q TargetQName* possono essere presenti molti report di attività e messaggi di risposta traceroute. *CorrelId* viene utilizzato per identificare i report di attività o un messaggio di risposta di trace - route, correlato a un messaggio di trace - route. Specificare l'identificativo del messaggio di traccia - instradamento originale in *CorrelId*.

Il formato di *CorrelId* è una stringa esadecimale di 48 caratteri.

I seguenti parametri possono essere utilizzati quando si visualizzano le informazioni sull'attività accumulate precedentemente o quando si visualizzano le informazioni sull'attività corrente per un messaggio di instradamento traccia:

-b

Specifica che l'applicazione di instradamento di visualizzazione IBM MQ visualizzerà solo i report di attività o un messaggio di risposta traceroute relativo ad un messaggio. Ciò consente di visualizzare nuovamente le informazioni sull'attività in un momento successivo.

Se non si specifica questo parametro, l'applicazione di instradamento di visualizzazione IBM MQ otterrà in modo distruttivo i report di attività o un messaggio di risposta di instradamento di traccia correlato a un messaggio.

-v summary | all | none | outline DisplayOption

riepilogo

Vengono visualizzate le code attraverso le quali è stato instradato il messaggio di trace - route.

tutti

Tutte le informazioni disponibili vengono visualizzate.

Nessuna

Non vengono visualizzate informazioni.

struttura DisplayOption

Specifica le opzioni di visualizzazione per il messaggio di trace - route. È possibile specificare più opzioni di visualizzazione utilizzando una virgola come separatore.

Se non viene fornito alcun valore, viene visualizzato quanto segue:

- Il nome dell'applicazione
- Il tipo di ciascuna operazione
- Tutti i parametri specifici dell'operazione

I valori possibili per *DisplayOption* sono:

attività

Vengono visualizzati tutti i parametri del gruppo non - PCF nei gruppi PCF *Attività* .

identificatori

Vengono visualizzati i valori con gli identificativi di parametro MQBACF_MSG_ID o MQBACF_CORREL_ID. Ciò sovrascrive *msgdelta*.

messaggio

Vengono visualizzati tutti i parametri del gruppo non PCF nei gruppi PCF *Messaggio* . Quando viene specificato questo valore, è impossibile specificare *msgdelta*.

delta msg

Vengono visualizzati tutti i parametri del gruppo non PCF nei gruppi PCF *Messaggio* , modificati dall'ultima operazione. Quando questo valore viene specificato, è possibile specificare *messaggio*.

operazione

Vengono visualizzati tutti i parametri del gruppo non PCF nei gruppi PCF *Operazione* .

traceroute

Vengono visualizzati tutti i parametri del gruppo non - PCF nei gruppi PCF *TraceRoute* .

Se non si specifica questo parametro, viene visualizzato un riepilogo dell'instradamento del messaggio.

Visualizzazione di informazioni aggiuntive

Quando un messaggio di traceroute viene instradato attraverso una rete di gestori code, le applicazioni scritte dall'utente possono registrare ulteriori informazioni scrivendo uno o più parametri PCF aggiuntivi nei dati del messaggio di traceroute o nei dati del messaggio di un report di attività. Affinché l'applicazione di instradamento di visualizzazione IBM MQ visualizzi ulteriori informazioni in un formato leggibile, è necessario che siano registrate in uno specifico formato, come descritto in ["Ulteriori informazioni sull'attività"](#) a pagina 79.

IBM MQ visualizza esempi di applicazione di instradamento

I seguenti esempi mostrano come utilizzare l'applicazione di instradamento di visualizzazione IBM MQ . In ogni esempio, due gestori code (QM1 e QM2) sono interconnessi da due canali (QM2.TO.QM1 e QM1.TO.QM2).

Esempio 1 - Richiesta di report di attività

Visualizzare le informazioni sull'attività da un messaggio di instradamento traccia consegnato alla coda di destinazione

In questo esempio, l'applicazione di instradamento di visualizzazione IBM MQ si connette al gestore code QM1 e viene utilizzata per generare e consegnare un messaggio di instradamento traccia alla coda di destinazione, TARGET.Q, sul gestore code remoto, QM2. L'opzione di prospetto necessaria viene specificata in modo che i prospetti di attività vengano richiesti quando il messaggio di risposta di traccia - instradamento viene instradato. All'arrivo nella coda di destinazione, il messaggio di trace - route viene eliminato. Le informazioni sull'attività restituite all'applicazione di instradamento di visualizzazione IBM MQ utilizzando i report di attività vengono messe in ordine e visualizzate.

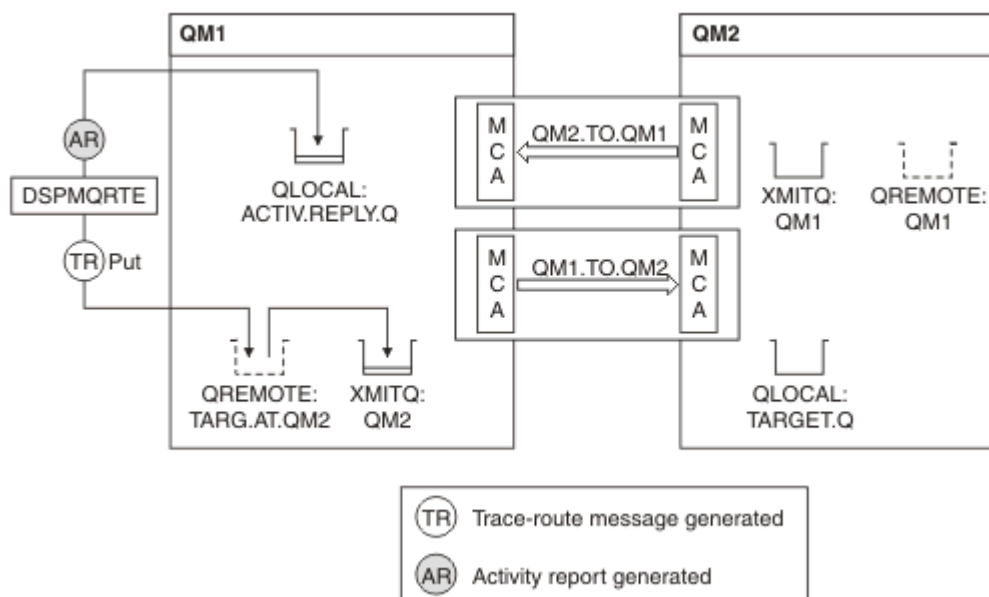


Figura 9. Richiesta di report di attività, diagramma 1

- L'attributo ACTIVREC di ciascun gestore code (QM1 e QM2) è impostato su MSG.
- Viene immesso il seguente comando:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1 è il nome del gestore code a cui si connette l'applicazione di instradamento di visualizzazione IBM MQ TARG.AT.QM2 è il nome della coda di destinazione e ACTIV.REPLY.Q è il nome della coda a cui è necessario inviare tutte le risposte al messaggio di indirizzamento traccia.

I valori predefiniti vengono assunti per tutte le opzioni che non sono specificate, ma si noti in modo particolare l'opzione -f (il messaggio trace - route viene inoltrato solo a un gestore code che rispetta il parametro Deliver del gruppo PCF TraceRoute), l'opzione -d (all'arrivo, il messaggio trace - route non viene inserito nella coda di destinazione), l'opzione -ro (sono specificate le opzioni del report MQRO_ACTIVITY e MQRO_DISCARD_MSG) e l'opzione -t (viene registrata l'attività del livello di dettaglio medio).

- DSPMQRTE genera il messaggio trace - route e lo inserisce nella coda remota TARG.AT.QM2.
- DSPMQRTE esamina quindi il valore dell'attributo ACTIVREC del gestore code QM1. Il valore è MSG, quindi DSPMQRTE genera un report di attività e lo inserisce sulla coda di risposta ACTIV.REPLY.Q.

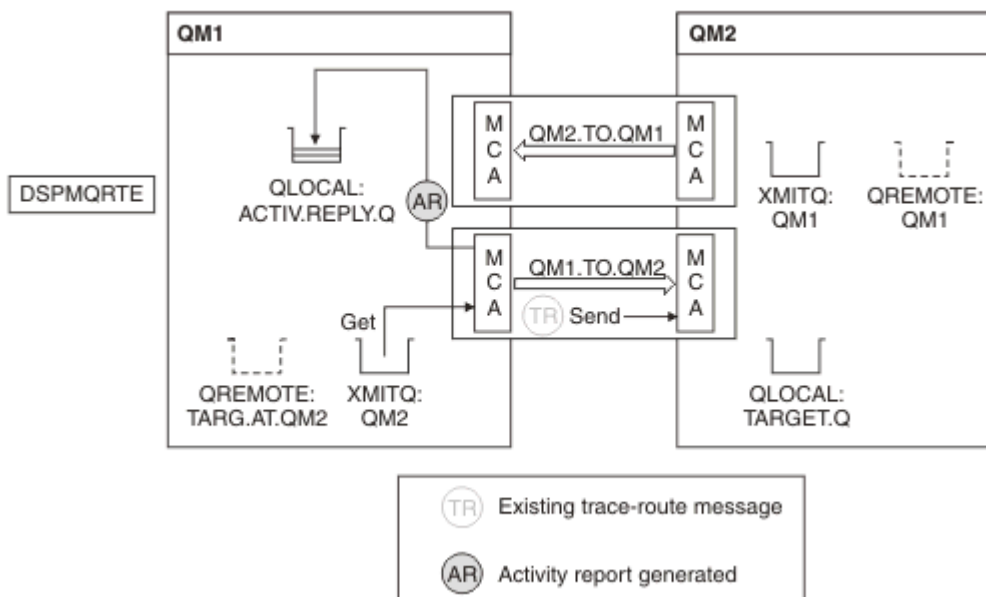


Figura 10. Richiesta di report di attività, diagramma 2

- L'agent MCA (message channel agent) di invio richiama il messaggio trace - route dalla coda di trasmissione. Il messaggio è un messaggio di instradamento traccia, quindi l'MCA inizia a registrare le informazioni sull'attività.
- L'attributo ACTIVREC del gestore code (QM1) è MSG e l'opzione MQRO_ACTIVITY è specificata nel campo Report del descrittore del messaggio, quindi l'MCA genererà successivamente un report di attività. il valore del parametro RecordedActivities nel gruppo PCF TraceRoute viene incrementato di 1.
- L'MCA verifica che il valore MaxActivities nel gruppo PCF TraceRoute non sia stato superato.
- Prima che il messaggio venga inoltrato a QM2, l'MCA segue l'algoritmo descritto in [Inoltro](#) (passi "1" a pagina 76, "4" a pagina 76 e "5" a pagina 77) e l'MCA sceglie di inviare il messaggio.
- MCA genera quindi un report di attività e lo inserisce nella coda di risposte (ACTIV.REPLY.Q).

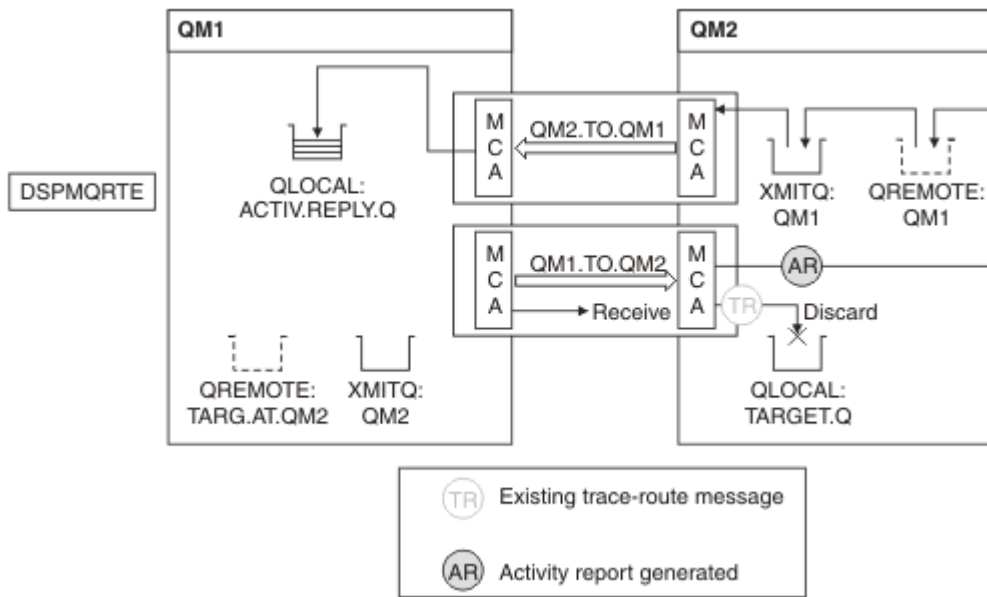


Figura 11. Richiesta di report di attività, diagramma 3

- L'MCA ricevente riceve il messaggio trace - route dal canale. Il messaggio è un messaggio di instradamento traccia, quindi l'MCA inizia a registrare le informazioni sull'attività.
- Se il gestore code da cui proviene il messaggio di traceroute è IBM WebSphere MQ 5.3.1 o precedente, l'MCA incrementa di 1 il parametro DiscontinuityCount del PCF TraceRoute . In questo caso non è così.
- L'attributo ACTIVREC del gestore code (QM2) è MSG e l'opzione MQRO_ACTIVITY è specificata, pertanto l'MCA genererà un report di attività. Il valore del parametro RecordedActivities viene incrementato di 1.
- La coda di destinazione è una coda locale, quindi il messaggio viene eliminato con il feedback MQFB_NOT_LIVER, in conformità con il valore del parametro Deliver nel gruppo PCF TraceRoute .
- L'MCA genera quindi il report di attività finale e lo inserisce nella coda di risposte. Ciò si risolve nella coda di trasmissione associata al gestore code QM1 e il report di attività viene restituito al gestore code QM1 (ACTIV.REPLY.Q).

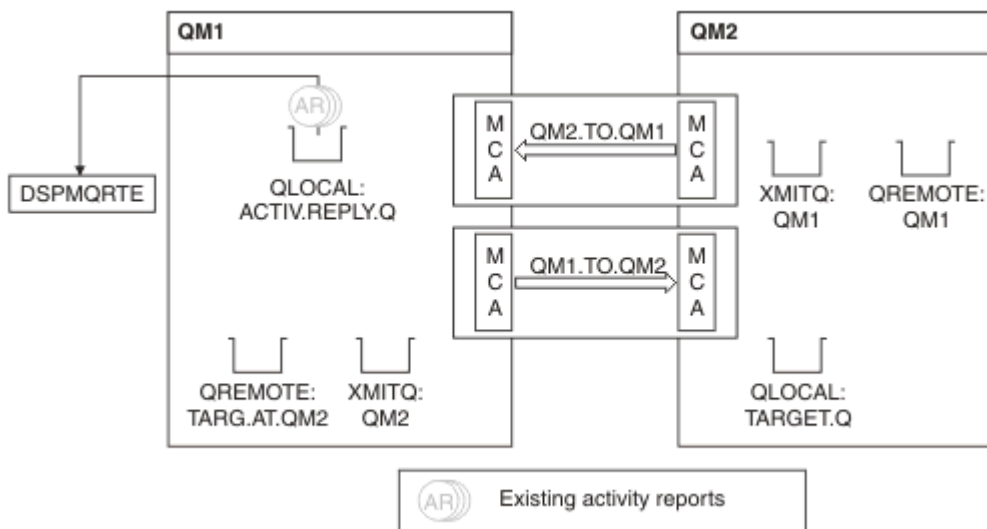


Figura 12. Richiesta di report di attività, diagramma 4

- Nel frattempo, DSPMQRTE ha eseguito continuamente MQGET sulla coda di risposta (ACTIV.REPLY.Q), in attesa di report di attività. Attenderà fino a 120 secondi (60 secondi in più rispetto alla scadenza del messaggio di trace - route) poiché -w non è stato specificato quando è stato avviato DSPMQRTE.
- DSPMQRTE richiama i 3 report di attività dalla coda di risposta.
- I report attività vengono ordinati utilizzando i parametri RecordedActivities, UnrecordedActivities e DiscontinuityCount nel gruppo PCF TraceRoute per ognuna delle attività. L'unico valore diverso da zero in questo esempio è RecordedActivities, quindi questo è l'unico parametro effettivamente utilizzato.
- Il programma termina non appena viene visualizzata l'operazione di eliminazione. Anche se l'operazione finale è stata un'eliminazione, viene considerata come se un'operazione di inserimento avesse luogo perché il feedback è MQFB_NOT_CONSEGNATO.

L'output visualizzato è il seguente:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq 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.
```

Esempio 2 - Richiesta di un messaggio di risposta trace - route

Genera e consegna un messaggio di instradamento traccia alla coda di destinazione

In questo esempio, l'applicazione di instradamento di visualizzazione IBM MQ si connette al gestore code QM1 e viene utilizzata per generare e consegnare un messaggio di instradamento traccia alla coda di destinazione, TARGET.Q, sul gestore code remoto, QM2. L'opzione necessaria viene specificata in modo che le informazioni sull'attività vengano accumulate nel messaggio di instradamento traccia. All'arrivo nella coda di destinazione, viene richiesto un messaggio di risposta di tracerout e il messaggio di tracerout viene eliminato.

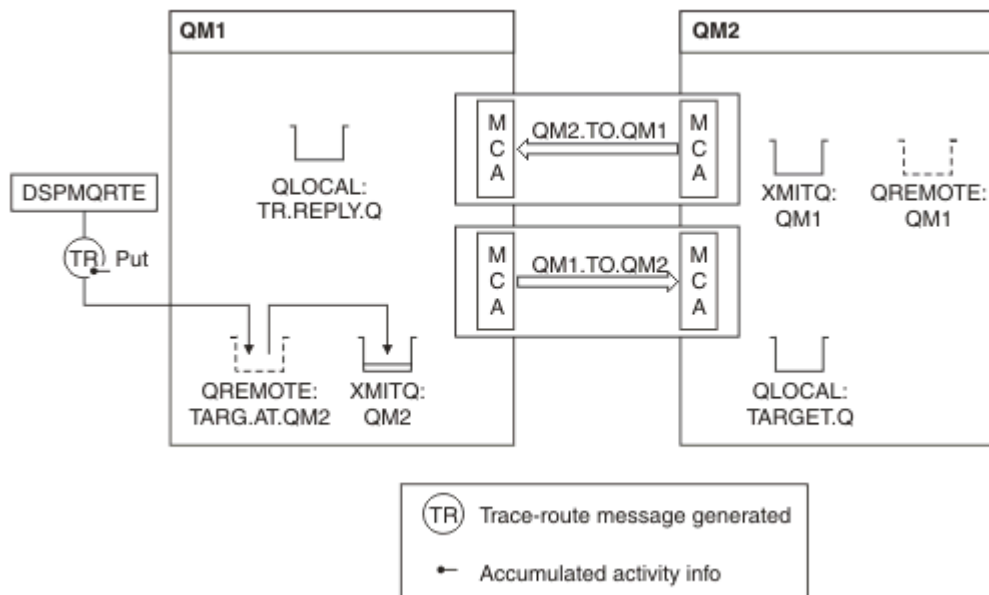


Figura 13. Richiesta di un messaggio di risposta di tracerout, diagramma 1

- l'attributo ROUTEREC di ciascun gestore code (QM1 e QM2) è impostato su MSG.
- Viene immesso il seguente comando:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq TR.REPLY.Q -ac -ar -ro discard
```

QM1 è il nome del gestore code a cui si connette l'applicazione di instradamento di visualizzazione IBM MQ TARG.AT.QM2 è il nome della coda di destinazione e ACTIV.REPLY.Q è il nome della coda a cui è necessario inviare tutte le risposte al messaggio di indirizzamento traccia. L'opzione -ac indica che le informazioni sull'attività vengono accumulate nel messaggio di trace - route, mentre l'opzione -ar specifica che tutta l'attività accumulata viene inviata alla coda reply - to specificata dall'opzione -rq (ovvero, TR.REPLY.Q). L'opzione -ro specifica che l'opzione di report MQRO_DISCARD_MSG è impostata, il che significa che i report di attività non vengono generati in questo esempio.

- DSPMQRTE accumula le informazioni sull'attività nel messaggio traceroute prima che il messaggio venga inserito nell'instradamento di destinazione. L'attributo del gestore code ROUTEREC non deve essere DISABLED perché questo accada.

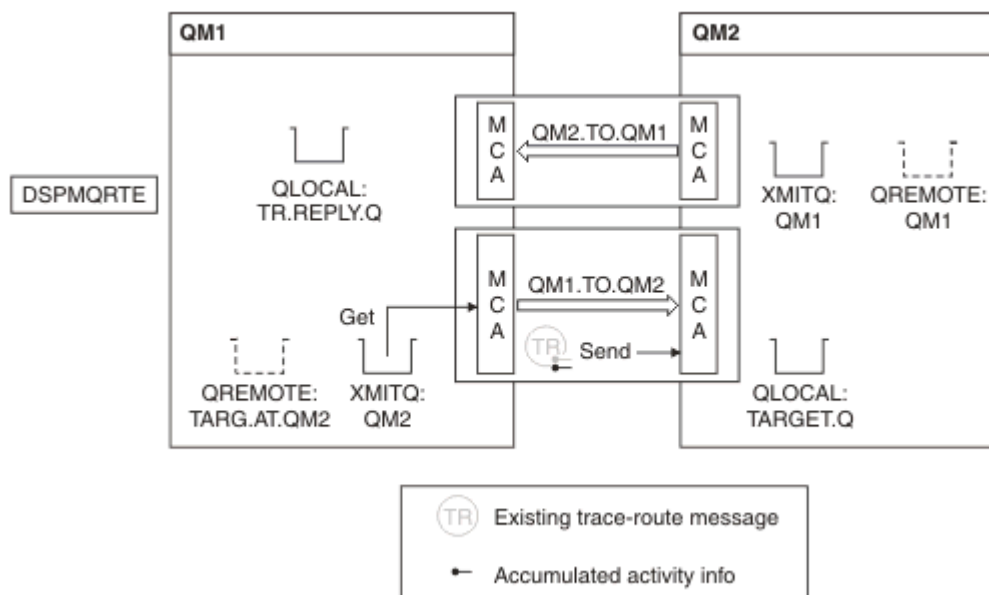


Figura 14. Richiesta di un messaggio di risposta di trace - route, diagramma 2

- Il messaggio è un messaggio di instradamento traccia, quindi l'MCA mittente inizia a registrare le informazioni sull'attività.
- L'attributo del gestore code ROUTEREC su QM1 non è DISABLED, pertanto l'MCA accumula le informazioni sull'attività all'interno del messaggio prima che il messaggio venga inoltrato al gestore code QM2.

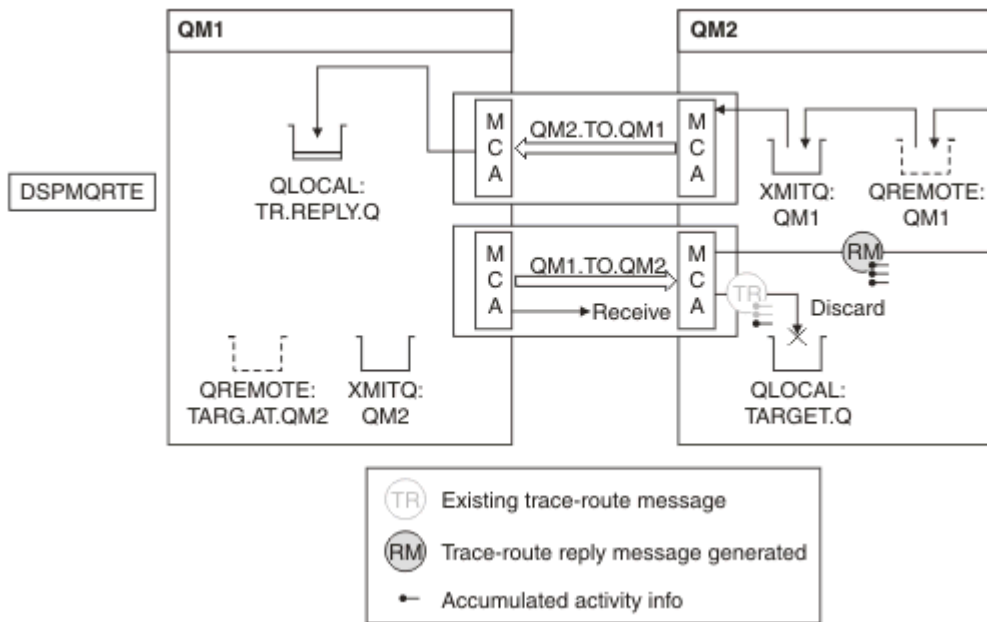


Figura 15. Richiesta di un messaggio di risposta di traceroute, diagramma 3

- Il messaggio è un messaggio di instradamento traccia, quindi l'MCA di ricezione inizia a registrare le informazioni sull'attività.
- L'attributo del gestore code ROUTEREC su QM2 non è DISABLED, quindi l'MCA accumula le informazioni all'interno del messaggio.
- La coda di destinazione è una coda locale, quindi il messaggio viene eliminato con il feedback MQFB_NOT_LIVER, in conformità con il valore del parametro Deliver nel gruppo PCF TraceRoute .
- Questa è l'ultima attività che si verificherà sul messaggio e, poiché l'attributo del gestore code ROUTEREC su QM1 non è DISABLED, l'MCA genera un messaggio di risposta traceroute in base al valore Accumulato. Il valore di ROUTEREC è MSG, quindi il messaggio di risposta viene inserito nella coda di risposta. Il messaggio di risposta contiene tutte le informazioni sull'attività accumulata dal messaggio di instradamento traccia.

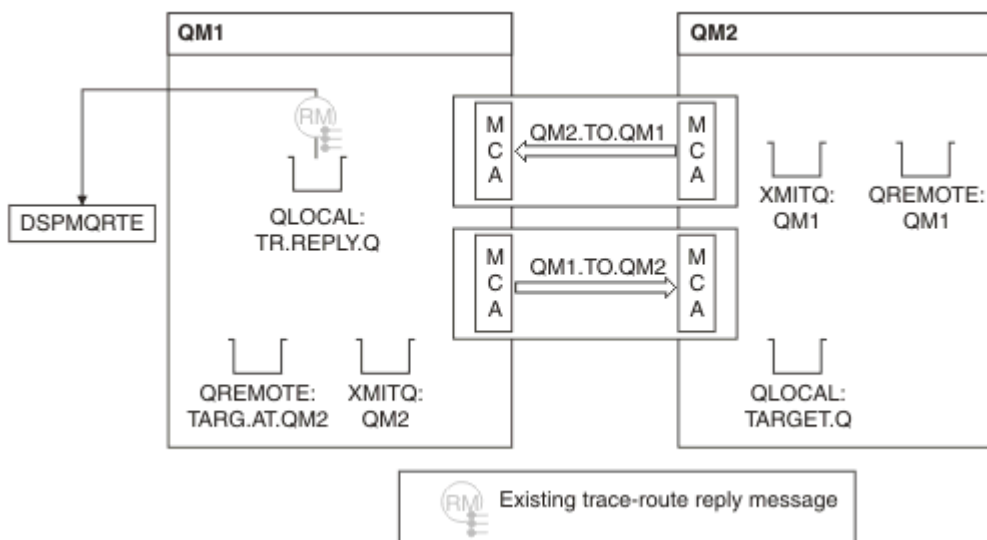


Figura 16. Richiesta di un messaggio di risposta di traceroute, diagramma 4

- Nel frattempo DSPMQRTE è in attesa che il messaggio di risposta traceroute ritorni alla coda di risposta. Quando viene restituito, DSPMQRTE analizza ogni attività che contiene e la stampa. L'operazione finale è un'operazione di eliminazione. DSPMQRTE termina dopo che è stato stampato.

L'output visualizzato è il seguente:

```
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.
```

Esempio 3 - Distribuzione di report di attività alla coda di sistema

Rilevare quando i report di attività vengono consegnati a code diverse dalla coda di risposta e utilizzare l'applicazione di instradamento di visualizzazione IBM MQ per leggere i report di attività dall'altra coda.

Questo esempio è uguale a “Esempio 1 - Richiesta di report di attività” a pagina 90, ad eccezione del fatto che QM2 ora ha il valore dell'attributo di gestione della coda ACTIVREC impostato su QUEUE. Canale QM1.TO.QM2 deve essere stato riavviato per rendere effettiva questa operazione.

Questo esempio dimostra come rilevare quando i report di attività vengono consegnati a code diverse dalla coda di risposta. Una volta rilevato, l'applicazione di instradamento di visualizzazione IBM MQ viene utilizzata per leggere i report di attività da un'altra coda.

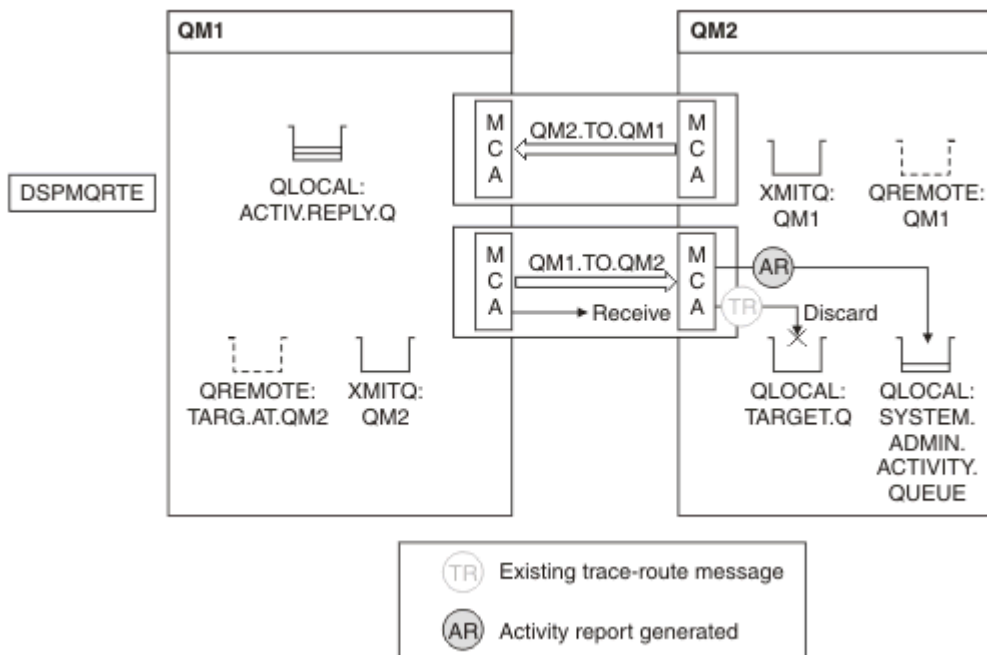


Figura 17. Consegna di report di attività alla coda di sistema, diagramma 1

- Il messaggio è un messaggio di instradamento traccia, quindi l'MCA di ricezione inizia a registrare le informazioni sull'attività.
- Il valore dell'attributo del gestore code ACTIVREC su QM2 è ora QUEUE, quindi l'MCA genera un report di attività, ma lo inserisce nella coda di sistema (SYSTEM.ADMIN.ACTIVITY.QUEUE) e non nella coda di risposta (ACTIV.REPLY.Q).

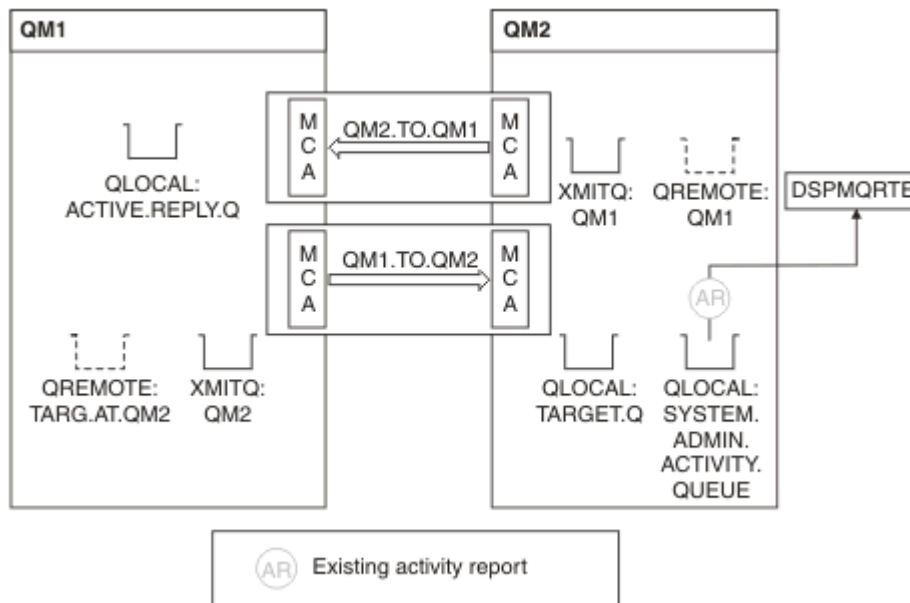


Figura 18. Consegna dei report di attività alla coda di sistema, diagramma 2

- Nel frattempo DSPMQRTE è in attesa che i report di attività arrivino su ACTIV.REPLY.Q. Solo due arrivano. DSPMQRTE continua ad attendere 120 secondi perché sembra che l'instradamento non sia stato ancora completato.

L'output visualizzato è il seguente:

```
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.
```

- L'ultima operazione osservata da DSPMQRTE è stata Invio, quindi il canale è in esecuzione. Ora dobbiamo capire perché non abbiamo ricevuto ulteriori report di attività dal gestore code QM2 (come identificato nel nome RemoteQMgr).
- Per controllare se sono presenti informazioni di attività sulla coda di sistema, avviare DSPMQRTE su QM2 per tentare di raccogliere ulteriori report di attività. Utilizzare il seguente comando per avviare DSPMQRTE:

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

dove 414D51204C41524745512020202020A3C9154220001502 è il MsgId del messaggio trace - route inserito.

- DSPMQRTE quindi esegue nuovamente una sequenza di MQGET, in attesa di risposte sulla coda di attività del sistema relative al messaggio di traceroute con l'identificativo specificato.

- DSPMQRTE ottiene un altro report di attività, che visualizza. DSPMQRTE determina che i precedenti report di attività sono mancanti e visualizza un messaggio che lo informa. Sappiamo già di questa parte del percorso, però.

L'output visualizzato è il seguente:

```

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.

```

- Questo report di attività indica che le informazioni sull'instradamento sono ora complete. Non si è verificato alcun problema.
- Solo perché le informazioni di instradamento non sono disponibili o perché DSPMQRTE non può visualizzare tutto l'instradamento, ciò non significa che il messaggio non è stato consegnato. Ad esempio, gli attributi del gestore code di gestori code differenti potrebbero essere diversi oppure una coda di risposta potrebbe non essere definita per ottenere la risposta.

Esempio 4 - Diagnosi di un problema di canale

Diagnosticare un problema in cui il messaggio di traccia - route non raggiunge la coda di destinazione

In questo esempio, l'applicazione di visualizzazione del percorso IBM MQ si connette al gestore code QM1, genera un messaggio di traccia del percorso e tenta di consegnarlo alla coda di destinazione TARGET.Q, sul gestore code remoto, QM2. In questo esempio il messaggio traccia - route non raggiunge la coda di destinazione. Il report di attività disponibile viene utilizzato per diagnosticare il problema.

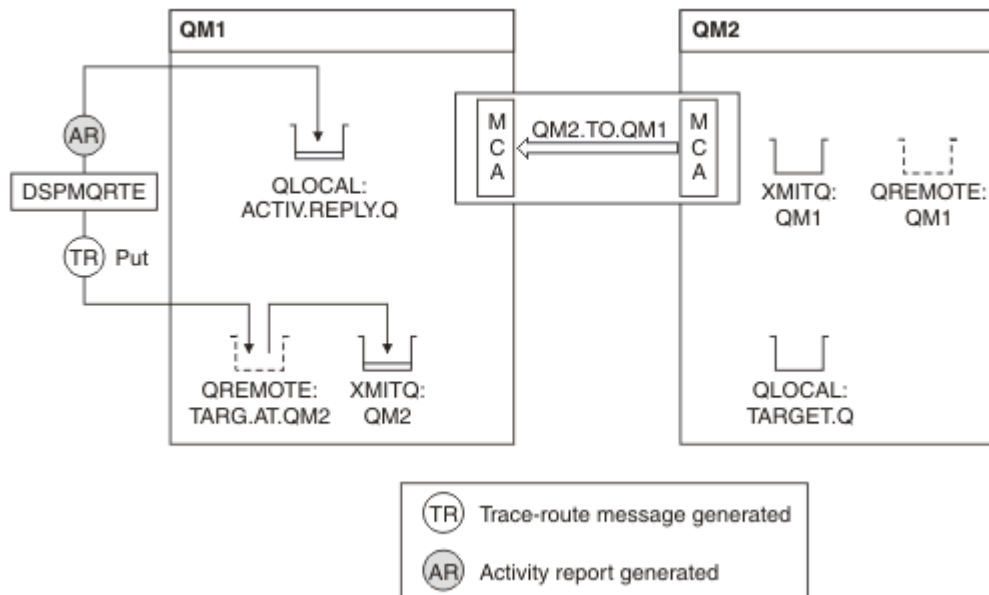


Figura 19. Diagnosi di un problema di canale

- In questo esempio, il canale QM1.TO.QM2 non è in esecuzione.
- DSPMQRTE inserisce un messaggio di trace - route (come nell'esempio 1) nella coda di destinazione e genera un report di attività.
- Non esiste alcun MCA per richiamare il messaggio dalla coda di trasmissione (QM2), quindi questo è l'unico report di attività che DSPMQRTE richiama dalla coda di risposta. Questa volta il fatto che l'instradamento non sia completo indica un problema. L'amministratore può utilizzare la coda di trasmissione trovata in ResolvedQName per esaminare il motivo per il quale la coda di trasmissione non viene servita.

L'output visualizzato è il seguente:

```
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.
```

Riferimento report attività

Utilizzare questa pagina per ottenere una panoramica del formato del messaggio del report di attività. I dati del messaggio del report di attività contengono i parametri che descrivono l'attività.

Formato report attività

I report di attività sono messaggi di report IBM MQ standard contenenti un descrittore di messaggi e dati di messaggi. I report di attività sono messaggi PCF generati dalle applicazioni che hanno eseguito un'attività per conto di un messaggio quando è stato instradato attraverso una rete di gestori code.

I report di attività contengono le seguenti informazioni:

Un descrittore di messaggi

Una struttura MQMD

Dati messaggio

Si compone di quanto segue:

- Un'intestazione PCF incorporata (MQEPH).
- Dati del messaggio del report di attività.

I dati del messaggio del report di attività sono costituiti dal gruppo PCF *Attività* e, se generati per un messaggio di traccia - instradamento, dal gruppo PCF *TraceRoute* .

Tabella 19 a pagina 102 mostra la struttura di questi report, inclusi i parametri restituiti solo in determinate condizioni.

Tabella 19. Formato report attività

Struttura MQMD	Struttura MQEPH intestazione PCF integrata	Dati del messaggio del report attività
Identificativo struttura Versione della struttura Opzioni di documentazione Tipo messaggio Ora di scadenza Feedback Codifica ID set di caratteri codificato Formato del messaggio Priorità Persistenza ID messaggio Identificativo di correlazione Conteggio backout Coda di risposta Gestore code di risposta Identificativo utente Token account Dati identità applicazione Tipo di applicazione Nome applicazione Data Put Ora Put Dati origine applicazione ID gruppo Numero sequenza messaggi Offset Indicatori di messaggio Lunghezza originale	Identificativo struttura Versione della struttura Lunghezza struttura Codifica ID set di caratteri codificato Formato del messaggio Indicatori Intestazione PCF (MQCFH) Tipo di struttura Lunghezza struttura Versione della struttura Identificativo comando Numero sequenza messaggi Opzioni di controllo Codice di completamento Codice di errore Conteggio parametro	Attività Nome applicazione attività Tipo di applicazione attività Descrizione attività Operazione Tipo di operazione Data operazione Ora operazione Messaggio Lunghezza messaggio MQMD ⁸ EmbeddedMQMD Nome del gestore code Nome gruppo di condivisione code Nome coda ¹ ² ³ ⁷ Nome coda risolta ² ³ ⁷ Nome coda remota ³ ⁷ Nome gestore code remoto ² ³ ⁴ ⁵ ⁷ Livello sottoscrizione ⁹ Identificativo sottoscrizione ⁹ Feedback ³ ¹⁰ Nome canale ⁴ ⁵ Tipo di canale ⁴ ⁵ Nome coda di trasmissione ⁵ TraceRoute ⁶ Dettaglio Attività registrate Attività non registrate Conteggio discontinuità Attività massime Accumula Distribuzione

Note:

1. Restituito per le operazioni Get e Browse.
2. Restituito per operazioni di eliminazione.
3. Restituito per le operazioni di inserimento, di inserimento risposta e di inserimento prospetto.
4. Restituito per operazioni di ricezione.

5. Restituito per le operazioni di invio.
6. Restituito per i messaggi di traccia - instradamento.
7. Non restituito per le operazioni di inserimento in un argomento, contenute nelle attività di pubblicazione.
8. Non restituito per le operazioni di pubblicazione escluse. Per le operazioni di pubblicazione e pubblicazione scartata, restituite contenenti un sottoinsieme di parametri.
9. Restituito per le operazioni di pubblicazione, pubblicazione eliminata e pubblicazione esclusa.
10. Restituito per le operazioni di pubblicazione scartate ed escluse.

MQMD del report di attività (descrittore del messaggio)

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQMD per un report di attività

StrucId

Identificativo struttura:

Tipo dati

MQCHAR4

Valore

ID_STRUC_MQMD.

Version

Numero di versione della struttura

Tipo dati

MQLONG

Valori

Copiato dal descrittore del messaggio originale. I possibili valori sono:

MQMD_VERSION_1

Struttura descrittore del messaggio Version-1 , supportata in tutti gli ambienti.

MQMD_VERSION_2

Struttura del descrittore di messaggi Version-2 , supportata nei seguenti ambienti:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

e tutti i IBM MQ MQI clients connessi a questi sistemi.

Report

Opzioni per ulteriori messaggi di report

Tipo dati

MQLONG

Valore

Se MQRO_PASS_DISCARD_AND_SCADENZA o MQRO_DISCARD_MSG sono stati specificati nel campo *Report* del descrittore del messaggio originale:

MQRO_DISCARD

Il report viene eliminato se non può essere consegnato alla coda di destinazione.

Altrimenti:

MQRO_NONE

Nessun report richiesto.

MsgType

Indica il tipo di messaggio

Tipo dati

MQLONG

Valore

REPORT MQMT

Expiry

Durata del messaggio di report

Tipo dati

MQLONG

Valore

Se il campo *Prospetto* nel descrittore del messaggio originale è specificato come MQRO_PASS_DISCARD_AND_SCADENZA, viene utilizzato il tempo di scadenza rimanente dal messaggio originale.

Altrimenti:

MQEI_UNLIMITED

Il report non ha un'ora di scadenza.

Feedback

Descrizione: Feedback o codice di errore.

Tipo di dati: MQLONG.

Valore: **MQFB_ATTIVITÀ**
Report attività.

Encoding

Descrizione: Codifica numerica dei dati del messaggio di report.

Tipo di dati: MQLONG.

Valore: MQENC_NATIVE.

CodedCharSetId

Descrizione: L'identificativo della serie di caratteri dei dati del messaggio di prospetto.

Tipo di dati: MQLONG.

Valore: Impostare come appropriato.

Format

Descrizione: Nome formato dei dati del messaggio di report

Tipo di dati: MQCHAR8.

Valore: **MQFMT_EMBEDDED_PCF**
Messaggio PCF incorporato.

Priority

Descrizione: Notificare la priorità del messaggio.

Tipo di dati: MQLONG.

Valore: Copiato dal descrittore del messaggio originale.

Persistence

Descrizione: Notificare la persistenza del messaggio.
Tipo di dati: MQLONG.
Valore: Copiato dal descrittore del messaggio originale.

MsgId

Descrizione: L'identificativo del messaggio.
Tipo di dati: MQBYTE24.
Valori: Se il campo *Prospecto* nel descrittore del messaggio originale è specificato come MQRO_PASS_MSG_ID, viene utilizzato l'identificativo del messaggio originale.
In caso contrario, il gestore code genererà un valore univoco.

CorrelId

Descrizione: Identificativo di correlazione.
Tipo di dati: MQBYTE24.
Valore: Se il campo *Report* nel descrittore del messaggio originale è specificato come MQRO_PASS_CORREL_ID, viene utilizzato l'identificativo di correlazione del messaggio originale.
In caso contrario, l'identificativo del messaggio viene copiato dal messaggio originale.

BackoutCount

Descrizione: Contatore backout.
Tipo di dati: MQLONG.
Valore: 0.

ReplyToQ

Descrizione: Nome della coda di risposte.
Tipo di dati: MQCHAR48.
Valori: Vuoto.

ReplyToQMgr

Descrizione: Nome del gestore code di risposte.
Tipo di dati: MQCHAR48.
Valore: Il nome del gestore code che ha generato il messaggio di report.

UserIdentifier

Descrizione: L'identificativo utente dell'applicazione che ha creato il messaggio di report.
Tipo di dati: MQCHAR12.
Valore: Copiato dal descrittore del messaggio originale.

AccountingToken

Descrizione: Token di account che consente ad un'applicazione di addebitare il lavoro eseguito come risultato del messaggio.
Tipo di dati: MQBYTE32.
Valore: Copiato dal descrittore del messaggio originale.

ApplIdentityData

Descrizione: Dati dell'applicazione relativi all'identità.
Tipo di dati: MQCHAR32.
Valori: Copiato dal descrittore del messaggio originale.

PutApplType

Descrizione: Tipo di applicazione che inserisce il messaggio di prospetto.
Tipo di dati: MQLONG.
Valore: **Gestore code MQAT**
Messaggio generato dal gestore code.

PutApplName

Descrizione: Nome dell'applicazione che ha inserito il messaggio di report.
Tipo di dati: MQCHAR28.
Valore: I primi 28 byte del nome del gestore code o il nome dell'MCA che ha generato il messaggio di report.

PutDate

Descrizione: Data in cui è stato inserito il messaggio.
Tipo di dati: MQCHAR8.
Valore: Come generato dal gestore code.

PutTime

Descrizione: L'ora in cui è stato inserito il messaggio.
Tipo di dati: MQCHAR8.
Valore: Come generato dal gestore code.

ApplOriginData

Descrizione: Dati di applicazione relativi all'origine.
Tipo di dati: MQCHAR4.
Valore: Vuoto.

Se la *versione* è MQMD_VERSION_2, sono presenti i seguenti campi aggiuntivi:

GroupId

Descrizione: Identifica a quale gruppo di messaggi o messaggio logico appartiene il messaggio fisico.
Tipo di dati: MQBYTE24.

Valore: Copiato dal descrittore del messaggio originale.

MsgSeqNumber

Descrizione: Numero di sequenza del messaggio logico all'interno del gruppo.

Tipo di dati: MQLONG.

Valore: Copiato dal descrittore del messaggio originale.

Offset

Descrizione: L'offset dei dati nel messaggio fisico dall'inizio del messaggio logico.

Tipo di dati: MQLONG.

Valore: Copiato dal descrittore del messaggio originale.

MsgFlags

Descrizione: Indicatori di messaggio che specificano gli attributi del messaggio o ne controllano l'elaborazione.

Tipo di dati: MQLONG.

Valore: Copiato dal descrittore del messaggio originale.

OriginalLength

Descrizione: Lunghezza del messaggio originale.

Tipo di dati: MQLONG.

Valore: Copiato dal descrittore del messaggio originale.

Report attività MQEPH (intestazione PCF integrata)

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQEPH per un report di attività

La struttura MQEPH contiene una descrizione delle informazioni PCF che accompagnano i dati del messaggio di un report di attività e i dati del messaggio dell'applicazione che lo seguono.

Per un report di attività, la struttura MQEPH contiene i valori seguenti:

StrucId

Descrizione: Identificatore struttura.

Tipo di dati: MQCHAR4.

Valore: ID_STRUC_MQEPH.

Version

Descrizione: Numero di versione della struttura.

Tipo di dati: MQLONG.

Valori: MQEPH_VERSION_1.

StrucLength

Descrizione: Lunghezza della struttura.

Tipo di dati: MQLONG.

Valore: Lunghezza totale della struttura che include le strutture del parametro PCF che la seguono.

Encoding

Descrizione:	Codifica numerica dei dati del messaggio che seguono l'ultima struttura del parametro PCF.
Tipo di dati:	MQLONG.
Valore:	Se i dati provenienti dai dati del messaggio dell'applicazione originale sono inclusi nel messaggio del report, il valore verrà copiato dal campo <i>Codifica</i> del descrittore del messaggio originale. Altrimenti, 0.

CodedCharSetId

Descrizione:	L'identificativo della serie di caratteri dei dati del messaggio che seguono l'ultima struttura del parametro PCF.
Tipo di dati:	MQLONG.
Valore:	Se i dati dei dati del messaggio dell'applicazione originale sono inclusi nel messaggio del report, il valore verrà copiato dal campo <i>CodedCharSetId</i> del descrittore del messaggio originale. Altrimenti, MQCCSI_UNDEFINED.

Format

Descrizione:	Nome formato dei dati del messaggio che seguono l'ultima struttura del parametro PCF.
Tipo di dati:	MQCHAR8.
Valore:	Se i dati provenienti dai dati del messaggio dell'applicazione originale sono inclusi nel messaggio del report, il valore verrà copiato dal campo <i>Formato</i> del descrittore del messaggio originale. Altrimenti, MQFMT_NONE.

Flags

Descrizione:	Indicatori che specificano gli attributi della struttura o ne controllano l'elaborazione.
Tipo di dati:	MQLONG.
Valore:	MQEPH_CCSID_EMBEDDED Specifica che la serie di caratteri dei parametri contenenti i dati carattere viene specificata singolarmente all'interno del campo <i>CodedCharSetId</i> in ogni struttura.

PCFHeader

Descrizione:	Intestazione formato comando programmabile
Tipo di dati:	MQCFH.
Valore:	Consultare “Report attività MQCFH (intestazione PCF)” a pagina 108.

Report attività MQCFH (intestazione PCF)

Utilizzare questa pagina per visualizzare i valori PCF contenuti nella struttura MQCFH per un report di attività

Per un report di attività, la struttura MQCFH contiene i valori seguenti:

Type

Descrizione: Tipo di struttura che identifica il contenuto del messaggio di report.
Tipo di dati: MQLONG.
Valore: **REPORT MQCFT**
Il messaggio è un report.

StrucLength

Descrizione: Lunghezza della struttura.
Tipo di dati: MQLONG.
Valore: **LUNGHEZZA_STRUTTURA_MQCFH_STRUCT**
Lunghezza in byte della struttura MQCFH.

Version

Descrizione: Numero di versione della struttura.
Tipo di dati: MQLONG.
Valori: MQCFH_VERSION_3

Command

Descrizione: Identificativo comando. Identifica la categoria del messaggio.
Tipo di dati: MQLONG.
Valori: **MQCMD_ATTIVITY_MSG**
Attività del messaggio.

MsgSeqNumber

Descrizione: Numero di sequenza del messaggio. Questo è il numero di sequenza del messaggio all'interno di un gruppo di messaggi correlati.
Tipo di dati: MQLONG.
Valori: 1.

Control

Descrizione: Opzioni di controllo.
Tipo di dati: MQLONG.
Valori: MQCFC_LAST.

CompCode

Descrizione: Codice di completamento.
Tipo di dati: MQLONG.
Valori: MQCC_OK.

Reason

Descrizione: Codice di errore che qualifica il codice di completamento.
Tipo di dati: MQLONG.

Valori: MQRC_NONE.

ParameterCount

Descrizione: Conteggio delle strutture di parametri. Questo è il numero di strutture di parametri che seguono la struttura MQCFH. Una struttura di gruppi (MQCFGR) e le relative strutture di parametri incluse vengono conteggiate come una sola struttura.

Tipo di dati: MQLONG.

Valori: 1 o superiore.

Dati del messaggio del report attività

Utilizzare questa pagina per visualizzare i parametri contenuti dal gruppo PCF *Attività* in un messaggio del prospetto attività. Alcuni parametri vengono restituiti solo quando sono state eseguite operazioni specifiche.

I dati del messaggio del report di attività sono costituiti dal gruppo PCF *Attività* e, se generati per un messaggio di traccia - instradamento, dal gruppo PCF *TraceRoute*. Il gruppo PCF *Attività* è descritto in dettaglio in questo argomento.

Alcuni parametri, descritti come Dati del messaggio del report dell'attività specifica dell'operazione, vengono restituiti solo quando sono state eseguite operazioni specifiche.

Per un report di attività, i dati del messaggio del report di attività contengono i seguenti parametri:

Activity

Descrizione: Parametri raggruppati che descrivono l'attività.

Identificativo: MQGACF_ACTIVITY.

Tipo di dati: MQCFGR.

Incluso nel gruppo PCF: Nessuna.

Parametri nel gruppo PCF: *ActivityApplName*
ActivityApplType
ActivityDescription
Operation
TraceRoute

Restituito: Sempre.

ActivityApplName

Descrizione: Il nome dell'applicazione che ha eseguito l'attività.

Identificativo: MQCACF_APPL_NAME.

Tipo di dati: MQCFST.

Incluso nel gruppo PCF: *Attività*.

Lunghezza massima: LUNGHEZZA_NOME_APPL_MQ.

Restituito: Sempre.

ActivityApplType

Descrizione: Tipo di applicazione che ha eseguito l'attività.

Identificativo: TIPO_APPL_MQI.
Tipo di dati: MQCFIN.
Incluso nel gruppo *Attività*.
PCF:
Restituito: Sempre.

ActivityDescription

Descrizione: Descrizione dell'attività eseguita dall'applicazione.
Identificativo: MQCACF_ACTIVITY_DESCRIPTION.
Tipo di dati: MQCFST.
Incluso nel gruppo *Attività*.
PCF:
Lunghezza massima: 64
Restituito: Sempre.

Operation

Descrizione: Parametri raggruppati che descrivono un'operazione dell'attività.
Identificativo: MQGACF_OPERATION.
Tipo di dati: MQCFGR.
Incluso nel gruppo *Attività*.
PCF:
Parametri nel gruppo PCF: *OperationType*
OperationDate
OperationTime
Message
QMgrName
QSGName

Nota: Ulteriori parametri vengono restituiti in questo gruppo a seconda del tipo di operazione. Questi parametri aggiuntivi sono descritti come Dati del messaggio del prospetto di attività specifici dell'operazione.

Restituito: Un gruppo PCF *Operazione* per operazione nell'attività.

OperationType

Descrizione: Tipo di operazione eseguita.
Identificativo: MQIACF_OPERATION_TYPE.
Tipo di dati: MQCFIN.
Incluso nel gruppo *Operazione*.
PCF:
Valori: MQOPER_*.
Restituito: Sempre.

OperationDate

Descrizione: Data in cui è stata eseguita l'operazione.
Identificativo: MQCACF_OPERATION_DATE.
Tipo di dati: MQCFST.
Incluso nel gruppo *Operazione*.
PCF:
Lunghezza massima: MQ_DATE_LENGTH.
Restituito: Sempre.

OperationTime

Descrizione: L'ora in cui è stata eseguita l'operazione.
Identificativo: MQCACF_OPERATION_TIME.
Tipo di dati: MQCFST.
Incluso nel gruppo *Operazione*.
PCF:
Lunghezza massima: MQ_TIME_LENGTH.
Restituito: Sempre.

Message

Descrizione: I parametri raggruppati che descrivono il messaggio che ha causato l'attività.
Identificativo: MQGACF_MESSAGE.
Tipo di dati: MQCFGR.
Incluso nel gruppo *Operazione*.
PCF:
Parametri nel gruppo: *MsgLength*
MQMD
EmbeddedMQMD
Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

MsgLength

Descrizione: La lunghezza del messaggio che ha causato l'attività, prima che si verificasse l'attività.
Identificativo: MQIACF_MSG_LENGTH.
Tipo di dati: MQCFIN.
Incluso nel gruppo *Messaggio*.
PCF:
Restituito: Sempre.

MQMD

Descrizione: Parametri raggruppati correlati al descrittore del messaggio che ha causato l'attività.

Identificativo: MQGACF_MQMD.

Tipo di dati: MQCFGR.

Incluso nel gruppo *Messaggio*.
PCF:

Parametri nel gruppo:

- 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*

Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

EmbeddedMQMD

Descrizione: Parametri raggruppati che descrivono il descrittore del messaggio integrato in un messaggio su una coda di trasmissione.

Identificativo: MQGACF_EMBEDDED_MQMD.

Tipo di dati: MQCFGR.

Incluso nel gruppo *Messaggio*.
PCF:

Parametri nel gruppo:	<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>
-----------------------	---

Restituito: Per le operazioni Get in cui la coda si risolve in una coda di trasmissione.

StrucId

Descrizione: Identificativo struttura

Identificativo: MQCACF_STRUC_ID.

Tipo di dati: MQCFST.

Incluso nel gruppo PCF: *MQMD* o *EmbeddedMQMD*.

Lunghezza massima: 4.

Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione e di pubblicazione scartate.

Version

Descrizione: Numero di versione della struttura.

Identificativo: MQIACF_VERSION.

Tipo di dati: MQCFIN.

Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.

PCF:

Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione e di pubblicazione scartate.

Report

Descrizione: Opzioni per i messaggi di report.

Identificativo: MQIACF_REPORT.

Tipo di dati: MQCFIN.

Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.

PCF:

Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione e di pubblicazione scartate.

MsgType

Descrizione: Indica il tipo di messaggio.

Identificativo: MQIACF_MSG_TYPE.

Tipo di dati: MQCFIN.

Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.

PCF:

Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione e di pubblicazione scartate.

Expiry

Descrizione: Durata del messaggio.

Identificativo: MQIACF_SCADENZA.

Tipo di dati: MQCFIN.

Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.

PCF:

Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione e di pubblicazione scartate.

Feedback

Descrizione: Feedback o codice di errore.

Identificativo: MQIACF_FEEDBACK.

Tipo di dati: MQCFIN.

Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.

PCF:

Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione e di pubblicazione scartate.

Encoding

Descrizione: Codifica numerica dei dati del messaggio.

Identificativo: MQIACF_ENCODING.

Tipo di dati: MQCFIN.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in *MQMD* per le operazioni di pubblicazione e di pubblicazione scartate.

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio.
Identificativo: MQIA_CODED_CHAR_SET_ID.
Tipo di dati: MQCFIN.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in *MQMD* per le operazioni di pubblicazione e di pubblicazione scartate.

Format

Descrizione: Nome formato dei dati di messaggio
Identificativo: MQCACH_FORMAT_NAME.
Tipo di dati: MQCFST.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Lunghezza massima: MQ_FORMAT_LENGTH.
Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

Priority

Descrizione: Priorità del messaggio.
Identificativo: MQIACF_PRIORITY.
Tipo di dati: MQCFIN.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

Persistence

Descrizione: Durata del messaggio.
Identificativo: MQIACF_PERSISTENCE.
Tipo di dati: MQCFIN.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

MsgId

Descrizione: L'identificativo del messaggio.

Identificativo: ID_MSG_MQBACF.
Tipo di dati: MQCFBS.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Lunghezza massima: MQ_MSG_ID_LENGTH.
Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

CorrelId

Descrizione: Identificativo di correlazione.
Identificativo: ID_CORREL_MQBACF.
Tipo di dati: MQCFBS.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Lunghezza massima: MQ_CORREL_ID_LENGTH.
Restituito: Sempre, tranne per le operazioni di pubblicazione escluse.

BackoutCount

Descrizione: Contatore backout.
Identificativo: MQIACF_BACKOUT_COUNT.
Tipo di dati: MQCFIN.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse e in *MQMD* per le operazioni di pubblicazione e di pubblicazione scartate.

ReplyToQ

Descrizione: Nome della coda di risposte.
Identificativo: MQCACF_REPLY_TO_Q.
Tipo di dati: MQCFST.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:
Lunghezza massima: LUNGHEZZA_NOME_Q_MQ.
Restituito: Sempre, tranne che per le operazioni di pubblicazione escluse e in *MQMD* per le operazioni di pubblicazione ed eliminazione.

ReplyToQMgr

Descrizione: Nome del gestore code di risposte.
Identificativo: MQCACF_REPLY_TO_Q_MGR.
Tipo di dati: MQCFST.
Incluso nel gruppo *MQMD* o *EmbeddedMQMD*.
PCF:

Lunghezza massima: MQ_Q_MGR_NAME_LENGTH.
Restituito: Sempre, tranne che per le operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

UserIdentifier

Descrizione: L'identificativo utente dell'applicazione che ha generato il messaggio.
Identificativo: MQCACF_USER_IDENTIFIER.
Tipo di dati: MQCFST.
Incluso nel gruppo PCF: *MQMD* o *EmbeddedMQMD*.
Lunghezza massima: MQ_USER_ID_LENGTH.
Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse.

AccountingToken

Descrizione: Token di account che consente ad un'applicazione di addebitare il lavoro eseguito come risultato del messaggio.
Identificativo: MQBACF_ACCOUNTING_TOKEN.
Tipo di dati: MQCFBS.
Incluso nel gruppo PCF: *MQMD* o *EmbeddedMQMD*.
Lunghezza massima: MQ_ACCOUNTING_TOKEN_LENGTH.
Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse.

ApplIdentityData

Descrizione: Dati dell'applicazione relativi all'identità.
Identificativo: MQCACF_APPL_IDENTITY_DATA.
Tipo di dati: MQCFST.
Incluso nel gruppo PCF: *MQMD* o *EmbeddedMQMD*.
Lunghezza massima: MQ_APPL_IDENTITY_DATA_LENGTH.
Restituito: Sempre, ad eccezione delle operazioni di pubblicazione escluse.

PutApplType

Descrizione: Tipo di applicazione che inserisce il messaggio.
Identificativo: TIPO_APPL_MQI.
Tipo di dati: MQCFIN.
Incluso nel gruppo PCF: *MQMD* o *EmbeddedMQMD*.
Restituito: Sempre, tranne che per le operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

PutApplName

Descrizione:	Nome dell'applicazione che inserisce il messaggio.
Identificativo:	MQCACF_APPL_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Lunghezza massima:	LUNGHEZZA_NOME_APPL_MQ.
Restituito:	Sempre, tranne che per le operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

PutDate

Descrizione:	Data in cui è stato inserito il messaggio.
Identificativo:	MQCACF_PUT_DATE.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Lunghezza massima:	LUNGHEZZA_DATA_PUT_MQ.
Restituito:	Sempre, tranne che per le operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

PutTime

Descrizione:	L'ora in cui è stato inserito il messaggio.
Identificativo:	MQCACF_PUT_TIME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Lunghezza massima:	MQ_PUT_TIME_LENGTH.
Restituito:	Sempre, tranne che per le operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

ApplOriginData

Descrizione:	Dati di applicazione relativi all'origine.
Identificativo:	MQCACF_APPL_ORIGIN_DATA.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Lunghezza massima:	MQ_APPL_ORIGIN_DATA_LENGTH.
Restituito:	Sempre, tranne che per le operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

GroupId

Descrizione:	Identifica a quale gruppo di messaggi o messaggio logico appartiene il messaggio fisico.
Identificativo:	ID_GROUP_MQBACF.
Tipo di dati:	MQCFBS.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Lunghezza massima:	LUNGHEZZA_ID_GRUPPO_MQ.
Restituito:	Se la <i>Versione</i> è specificata come <i>MQMD_VERSION_2</i> . Non restituito nelle operazioni di pubblicazione escluse e in <i>MQMD</i> per le operazioni di pubblicazione ed eliminazione.

MsgSeqNumber

Descrizione:	Numero di sequenza del messaggio logico all'interno del gruppo.
Identificativo:	MQIACH_MSG_SEQUENCE_NUMBER.
Tipo di dati:	MQCFIN.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Restituito:	Se <i>Versione</i> è specificato come <i>MQMD_VERSION_2</i> . Non restituito nelle operazioni di pubblicazione escluse e in <i>MQMD</i> per le operazioni di pubblicazione ed eliminazione.

Offset

Descrizione:	L'offset dei dati nel messaggio fisico dall'inizio del messaggio logico.
Identificativo:	MQIACF_OFFSET.
Tipo di dati:	MQCFIN.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Restituito:	Se <i>Versione</i> è specificato come <i>MQMD_VERSION_2</i> . Non restituito nelle operazioni di pubblicazione escluse e in <i>MQMD</i> per le operazioni di pubblicazione ed eliminazione.

MsgFlags

Descrizione:	Indicatori di messaggio che specificano gli attributi del messaggio o ne controllano l'elaborazione.
Identificativo:	MQIACF_MSG_FLAGS.
Tipo di dati:	MQCFIN.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Restituito:	Se <i>Versione</i> è specificato come <i>MQMD_VERSION_2</i> . Non restituito nelle operazioni di pubblicazione escluse e in <i>MQMD</i> per le operazioni di pubblicazione ed eliminazione.

OriginalLength

Descrizione:	Lunghezza del messaggio originale.
Identificativo:	MQIACF_ORIGINAL_LENGTH.
Tipo di dati:	MQCFIN.
Incluso nel gruppo PCF:	<i>MQMD</i> o <i>EmbeddedMQMD</i> .
Restituito:	Se <i>Versione</i> è specificato come MQMD_VERSION_2. Non restituito nelle operazioni di pubblicazione escluse e in MQMD per le operazioni di pubblicazione ed eliminazione.

QMGrName

Descrizione:	Il nome del gestore code in cui è stata eseguita l'attività.
Identificativo:	MQCA_Q_MGR_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione</i> .
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_
Restituito:	Sempre.

QSGName

Descrizione:	Il nome del gruppo di condivisione code a cui appartiene il gestore code a cui è stata eseguita l'attività.
Identificativo:	MQCA_QSG_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione</i> .
Lunghezza massima:	LUNGHEZZA_NOME_QSG_MQ
Restituito:	Se l'attività è stata eseguita su un gestore code IBM MQ for z/OS .

TraceRoute

Descrizione:	Parametri raggruppati che specificano gli attributi del messaggio trace - route.
Identificativo:	MQGACF_TRACE_ROUTE.
Tipo di dati:	MQCFGR.
Contenuto nel gruppo PCF:	<i>Attività</i> .

Parametri nel gruppo:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
-----------------------	---

Restituito: Se l'attività è stata eseguita per conto del messaggio di trace - route.

I valori dei parametri nel gruppo PCF *TraceRoute* sono quelli del messaggio di traccia - instradamento nel momento in cui è stato generato il report di attività.

Dati del messaggio del report di attività specifici dell'operazione

Utilizzare questa pagina per visualizzare i parametri PCF aggiuntivi che potrebbero essere restituiti nel gruppo PCF *Operazione* in un report di attività, in base al valore del parametro *OperationType*

I parametri aggiuntivi variano a seconda dei seguenti tipi di operazione:

Ottieni / Sfoglia (MQOPER_GET/MQOPER_BROWSE)

L'attività aggiuntiva riporta i parametri dei dati dei messaggi restituiti nel gruppo PCF *Operazione* per il tipo di operazione Get / Browse (MQOPER_GET/MQOPER_BROWSE) (è stato ricevuto o sfogliato un messaggio su una coda).

QName

Descrizione:	Il nome della coda aperta.
Identificativo:	MQCA_Q_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione</i> .
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ
Restituito:	Sempre.

ResolvedQName

Descrizione:	Il nome in cui si risolve la coda aperta.
Identificativo:	MQCACF_RESOLVED_Q_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione</i> .
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ
Restituito:	Sempre.

Eliminazione (MQOPER_DISCARD)

L'attività aggiuntiva riporta i parametri dei dati dei messaggi restituiti nel gruppo PCF *Operation* per il tipo di operazione Discard (MQOPER_DISCARD) (un messaggio è stato eliminato).

Feedback

Descrizione:	Il motivo per cui il messaggio è stato eliminato.
--------------	---

Identificativo: MQIACF_FEEDBACK.
Tipo di dati: MQCFIN.
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Sempre.

QName

Descrizione: Il nome della coda aperta.
Identificativo: MQCA_Q_NAME.
Tipo di dati: MQCFST.
Lunghezza massima: LUNGHEZZA_NOME_Q_MQ
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Se il messaggio è stato eliminato perché non è stato inserito correttamente in una coda.

RemoteQMgrName

Descrizione: Il nome del gestore code a cui era destinato il messaggio.
Identificativo: MQCA_REMOTE_Q_MGR_NAME.
Tipo di dati: MQCFST.
Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Se il valore di *Feedback* è MQFB_NOT_FORWARDED.

Pubblicazione / Pubblicazione eliminata / Pubblicazione esclusa (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)

I parametri aggiuntivi dei dati del messaggio del report di attività restituiti nel gruppo PCF *Operazione* per il tipo di operazione *Pubblicazione / Pubblicazione scartata / Pubblicazione esclusa (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)* (un messaggio di pubblicazione / sottoscrizione è stato recapitato, eliminato o escluso).

SubId

Descrizione: L'identificativo della sottoscrizione.
Identificativo: MQBACF_SUB_ID.
Tipo di dati: MQCFBS.
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Sempre.

SubLevel

Descrizione: Il livello di sottoscrizione.
Identificativo: MQIACF_SUB_LEVEL.

Tipo di dati: MQCFIN.
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Sempre.

Feedback

Descrizione: Il motivo dell'eliminazione del messaggio.
Identificativo: MQIACF_FEEDBACK.
Tipo di dati: MQCFIN.
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Se il messaggio è stato eliminato perché non è stato consegnato ad un sottoscrittore o perché il sottoscrittore è stato escluso.

L'operazione di pubblicazione MQOPER_PUBLISH fornisce informazioni su un messaggio consegnato a uno specifico sottoscrittore. Questa operazione descrive gli elementi del messaggio successivo che potrebbero essere stati modificati rispetto al messaggio descritto nell'operazione Put associata. Allo stesso modo di un'operazione Put, contiene un gruppo di messaggi MQGACF_MESSAGE e, al suo interno, un gruppo MQMD MQGACF_MQMD. Tuttavia, questo gruppo MQMD contiene solo i seguenti campi, che possono essere sovrascritti da un sottoscrittore: *Format, Priority, Persistence, MsgId, CorrelId, UserIdentifier, AccountingToken, ApplIdentityData*.

L' *SubId* e il *SubLevel* del sottoscrittore sono inclusi nelle informazioni sull'operazione. È possibile utilizzare *SubID* con il comando MQCMD_INQUIRE_SUBSCRIBER PCF per richiamare tutti gli altri attributi per un sottoscrittore.

L'operazione di pubblicazione MQOPER_DISCARDED_PUBLISH è analoga all'operazione di eliminazione utilizzata quando un messaggio non viene consegnato nella messaggistica point - to - point. Un messaggio non viene consegnato a un sottoscrittore se il messaggio è stato esplicitamente richiesto di non essere consegnato a una destinazione locale e questo sottoscrittore specifica una destinazione locale. Un messaggio viene inoltre considerato non consegnato se si verifica un problema nel richiamare il messaggio nella coda di destinazione, ad esempio, perché la coda è piena.

Le informazioni in un'operazione di pubblicazione scartata sono le stesse di un'operazione di pubblicazione, con l'aggiunta di un campo *Feedback* che fornisce i motivi per cui il messaggio non è stato consegnato. Questo campo di feedback contiene i valori MQFB_* o MQRC_* comuni con l'operazione MQOPER_DISCARD. Il motivo per cui si elimina una pubblicazione, invece di escluderlo, è lo stesso motivo per cui si elimina una pubblicazione.

L'operazione di pubblicazione esclusa MQOPER_EXCLUDED_PUBLISH fornisce informazioni su un sottoscrittore (subscriber) che è stato considerato per la consegna del messaggio, poiché l'argomento su cui il sottoscrittore (subscriber) sta effettuando la sottoscrizione corrisponde a quella dell'operazione Put associata, ma il messaggio non è stato consegnato al sottoscrittore (subscriber) in quanto altri criteri di selezione non corrispondono al messaggio che viene inserito nell'argomento. Come con un'operazione di pubblicazione scartata, il campo *Feedback* fornisce informazioni sul motivo per cui questa sottoscrizione è stata esclusa. Tuttavia, a differenza dell'operazione di pubblicazione scartata, non vengono fornite informazioni relative al messaggio poiché non è stato generato alcun messaggio per questo sottoscrittore.

Report inserimento / inserimento risposta / inserimento (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT)

L'attività aggiuntiva riporta i parametri dei dati dei messaggi restituiti nel gruppo PCF *Operazione* per il tipo di operazione del report Put / Put Reply / Put (MQOPER_PUT/MQOPER_PUT_REPLY/

MQOPER_PUT_REPORT) (un messaggio, un messaggio di risposta o un messaggio di report è stato inserito in una coda).

QName

Descrizione:	Il nome della coda aperta.
Identificativo:	MQCA_Q_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione.</i>
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ
Restituito:	Sempre, a parte un'eccezione: non restituito se l'operazione Put è su un argomento, contenuto in un'attività di pubblicazione.

ResolvedQName

Descrizione:	Il nome in cui si risolve la coda aperta.
Identificativo:	MQCACF_RESOLVED_Q_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione.</i>
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ
Restituito:	Quando è stato possibile risolvere la coda aperta. Non viene restituito se l'operazione Put è su un argomento, contenuto all'interno di un'attività di pubblicazione.

RemoteQName

Descrizione:	Il nome della coda aperta, come è noto sul gestore code remoto.
Identificativo:	MQCA_REMOTE_Q_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione.</i>
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ
Restituito:	Se la coda aperta è una coda remota. Non viene restituito se l'operazione Put è su un argomento, contenuto all'interno di un'attività di pubblicazione.

RemoteQMgrName

Descrizione:	Il nome del gestore code remoto su cui è definita la coda remota.
Identificativo:	MQCA_REMOTE_Q_MGR_NAME.
Tipo di dati:	MQCFST.
Incluso nel gruppo PCF:	<i>Operazione.</i>
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_

Restituito: Se la coda aperta è una coda remota. Non viene restituito se l'operazione Put è su un argomento, contenuto all'interno di un'attività di pubblicazione.

TopicString

Descrizione: La stringa di argomenti completa in cui viene inserito il messaggio.

Identificativo: MQCA_TOPIC_STRING.

Tipo di dati: MQCFST.

Incluso nel gruppo *Operazione*.

PCF:

Restituito: Se l'operazione Put è su un argomento, contenuto all'interno di un'attività di pubblicazione.

Feedback

Descrizione: Il motivo per cui il messaggio viene inserito nella coda di messaggi non instradabili.

Identificativo: MQIACF_FEEDBACK.

Tipo di dati: MQCFIN.

Incluso nel gruppo *Operazione*.

PCF:

Restituito: Se il messaggio è stato inserito nella coda di messaggi non recapitabili.

Ricezione (MQOPER_RECEIVE)

I parametri aggiuntivi dei dati del messaggio di report di attività restituiti nel gruppo PCF *Operazione* per il tipo di operazione di ricezione (MQOPER_RECEIVE) (è stato ricevuto un messaggio su un canale).

ChannelName

Descrizione: Il nome del canale su cui è stato ricevuto il messaggio.

Identificativo: MQCACH_CHANNEL_NAME.

Tipo di dati: MQCFST.

Incluso nel gruppo *Operazione*.

PCF:

Lunghezza massima: LUNGHEZZA_NOME_CANALE_MQ

Restituito: Sempre.

ChannelType

Descrizione: Il tipo di canale su cui è stato ricevuto il messaggio.

Identificativo: MQIACH_CHANNEL_TYPE.

Tipo di dati: MQCFIN.

Incluso nel gruppo *Operazione*.

PCF:

Restituito: Sempre.

RemoteQMgrName

Descrizione: Il nome del gestore code da cui è stato ricevuto il messaggio.

Identificativo: MQCA_REMOTE_Q_MGR_NAME.
Tipo di dati: MQCFST.
Incluso nel gruppo *Operazione*.
PCF:
Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_
Restituito: Sempre.

Invia (MQOPER_SEND)

I parametri aggiuntivi dei dati del messaggio di report di attività restituiti nel gruppo PCF *Operazione* per il tipo di operazione di invio (MQOPER_SEND) (un messaggio è stato inviato su un canale).

ChannelName

Descrizione: Il nome del canale a cui è stato inviato il messaggio.
Identificativo: MQCACH_CHANNEL_NAME.
Tipo di dati: MQCFST.
Incluso nel gruppo *Operazione*.
PCF:
Lunghezza massima: LUNGHEZZA_NOME_CANALE_MQ.
Restituito: Sempre.

ChannelType

Descrizione: Il tipo di canale a cui è stato inviato il messaggio.
Identificativo: MQIACH_CHANNEL_TYPE.
Tipo di dati: MQCFIN.
Incluso nel gruppo *Operazione*.
PCF:
Restituito: Sempre.

XmitQName

Descrizione: La coda di trasmissione dalla quale è stato richiamato il messaggio.
Identificativo: MQCACH_XMIT_Q_NAME.
Tipo di dati: MQCFST.
Incluso nel gruppo *Operazione*.
PCF:
Lunghezza massima: LUNGHEZZA_NOME_Q_MQ.
Restituito: Sempre.

RemoteQMgrName

Descrizione: Il nome del gestore code remoto a cui è stato inviato il messaggio.
Identificativo: MQCA_REMOTE_Q_MGR_NAME.
Tipo di dati: MQCFST.

Incluso nel gruppo *Operazione*.

PCF:

Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_

Restituito: Sempre.

Riferimento messaggio di traccia - instradamento

Utilizzare questa pagina per ottenere una panoramica del formato del messaggio di traccia - route. I dati del messaggio di traccia - instradamento includono parametri che descrivono le attività che il messaggio di traccia - instradamento ha causato

Formato del messaggio di tracerouto

I messaggi di traccia - instradamento sono messaggi IBM MQ standard che contengono un descrittore di messaggi e dati di messaggi. I dati del messaggio contengono informazioni sulle attività eseguite su un messaggio di traccia - route come è stato instradato attraverso una rete di gestori code.

I messaggi di traccia - route contengono le seguenti informazioni:

Un descrittore di messaggi

Una struttura MQMD, con il campo *Formato* impostato su MQFMT_ADMIN o MQFMT_EMBEDDED_PCF.

Dati messaggio

È composto da:

- Un'intestazione PCF (MQCFH) e i dati del messaggio di traccia - route, se *Formato* è impostato su MQFMT_ADMIN oppure
- Un'intestazione PCF incorporata (MQEPH), i dati del messaggio di instradamento della traccia e ulteriori dati del messaggio specificati dall'utente, se *Formato* è impostato su MQFMT_EMBEDDED_PCF.

Quando si utilizza l'applicazione di instradamento di visualizzazione IBM MQ per creare un messaggio di instradamento traccia, *Formato* è impostato su MQFMT_ADMIN.

Il contenuto dei dati del messaggio di traccia - route è determinato dal parametro *Accumulate* del gruppo PCF *TraceRoute* , come segue:

- Se *Accumulate* è impostato su MQROUTE_ACCUMULATE_NONE, i dati del messaggio di traccia - instradamento contengono il gruppo PCF *TraceRoute* .
- Se *Accumulate* è impostato su MQROUTE_ACCUMULATE_IN_MSG o MQROUTE_ACCUMULATE_AND_REPLY, i dati del messaggio di traccia - instradamento contengono il gruppo PCF *TraceRoute* e zero o più gruppi PCF *Activity* .

La [Tabella 20 a pagina 129](#) mostra la struttura di un messaggio di tracerouto.

Tabella 20. Formato del messaggio di traceroute

Struttura MQMD	Struttura MQEPH intestazione PCF integrata	Traccia - dati del messaggio di instradamento
Identificativo struttura Versione della struttura Opzioni di documentazione Tipo messaggio Ora di scadenza Feedback Codifica ID set di caratteri codificato Formato del messaggio Priorità Persistenza ID messaggio Identificativo di correlazione Conteggio backout Coda di risposta Gestore code di risposta Identificativo utente Token account Dati identità applicazione Tipo di applicazione Nome applicazione Data Put Ora Put Dati origine applicazione ID gruppo Numero sequenza messaggi Offset Indicatori di messaggio Lunghezza originale	Identificativo struttura Versione della struttura Lunghezza struttura Codifica ID set di caratteri codificato Formato del messaggio Indicatori Intestazione PCF (MQCFH) Tipo di struttura Lunghezza struttura Versione della struttura Identificativo comando Numero sequenza messaggi Opzioni di controllo Codice di completamento Codice di errore Conteggio parametro	TraceRoute Dettaglio Attività registrate Attività non registrate Conteggio discontinuità Attività massime Accumula Distribuzione

Traccia - instradamento messaggio MQMD (descrittore messaggio)

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQMD per un messaggio traceroute

StrucId

Descrizione: Identificatore struttura.
 Tipo di dati: MQCHAR4.
 Valore: ID_STRUC_MQMD.

Version

Descrizione: Numero di versione della struttura.
 Tipo di dati: MQLONG.
 Valori: **MQMD_VERSION_1.**

Report

Descrizione: Opzioni per i messaggi di report.

Tipo di dati: MQLONG.
Valore: Impostare in base ai requisiti. Di seguito sono riportate le seguenti opzioni di report comuni:

MQRO_DISCARD_MSG

Il messaggio viene eliminato all'arrivo in una coda locale.

MQRO_PASS_DISCARD_E_SCADENZA

Ogni risposta (report di attività o messaggio di risposta trace - route) avrà l'opzione di report MQRO_DISCARD_MSG impostata e la restante scadenza passata. Ciò garantisce che le risposte non restino nella rete del gestore code per un periodo di tempo indefinito.

MsgType

Descrizione: Tipo di messaggio.
Tipo di dati: MQLONG.
Valore: Se il parametro *Accumulate* nel gruppo TraceRoute è specificato come MQROUTE_ACCUMULATE_AND_REPLY, il tipo di messaggio è MQMT_REQUEST
Altrimenti:
MQMT_DATAGRAM.

Expiry

Descrizione: Durata del messaggio.
Tipo di dati: MQLONG.
Valore: Impostare in base ai requisiti. Questo parametro può essere utilizzato per garantire che i messaggi di traceroute non vengano lasciati in una rete di gestori code per un tempo indefinito.

Feedback

Descrizione: Feedback o codice di errore.
Tipo di dati: MQLONG.
Valore: **MQFB_NONE.**

Encoding

Descrizione: Codifica numerica dei dati del messaggio.
Tipo di dati: MQLONG.
Valore: Impostare come appropriato.

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio.
Tipo di dati: MQLONG.
Valore: Impostare come appropriato.

Format

Descrizione: Nome formato dei dati di messaggio
Tipo di dati: MQCHAR8.

Valore: **MMQFMT_ADMIN**
Messaggio di amministrazione. Nessun dato utente segue il gruppo PCF *TraceRoute* .

MQFMT_EMBEDDED_PCF
Messaggio PCF incorporato. I dati utente seguono il gruppo *TraceRoute* PCF.

Priority

Descrizione: Priorità del messaggio.
Tipo di dati: MQLONG.
Valore: Impostare in base ai requisiti.

Persistence

Descrizione: Durata del messaggio.
Tipo di dati: MQLONG.
Valore: Impostare in base ai requisiti.

MsgId

Descrizione: L'identificativo del messaggio.
Tipo di dati: MQBYTE24.
Valore: Impostare in base ai requisiti.

CorrelId

Descrizione: Identificativo di correlazione.
Tipo di dati: MQBYTE24.
Valore: Impostare in base ai requisiti.

BackoutCount

Descrizione: Contatore backout.
Tipo di dati: MQLONG.
Valore: 0.

ReplyToQ

Descrizione: Nome della coda di risposte.
Tipo di dati: MQCHAR48.
Valori: Impostare in base ai requisiti.

Se *MsgType* è impostato su MQMT_REQUEST o se per *Report* sono impostate opzioni di generazione report, questo parametro non deve essere vuoto.

ReplyToQMgr

Descrizione: Nome del gestore code di risposte.
Tipo di dati: MQCHAR48.
Valore: Impostare in base ai requisiti.

UserIdentifier

Descrizione: L'identificativo utente dell'applicazione che ha generato il messaggio.
Tipo di dati: MQCHAR12.
Valore: Impostare come normale.

AccountingToken

Descrizione: Token di account che consente ad un'applicazione di addebitare il lavoro eseguito come risultato del messaggio.
Tipo di dati: MQBYTE32.
Valore: Impostare come normale.

ApplIdentityData

Descrizione: Dati dell'applicazione relativi all'identità.
Tipo di dati: MQCHAR32.
Valori: Impostare come normale.

PutApplType

Descrizione: Tipo di applicazione che inserisce il messaggio.
Tipo di dati: MQLONG.
Valore: Impostare come normale.

PutApplName

Descrizione: Nome dell'applicazione che inserisce il messaggio.
Tipo di dati: MQCHAR28.
Valore: Impostare come normale.

PutDate

Descrizione: Data in cui è stato inserito il messaggio.
Tipo di dati: MQCHAR8.
Valore: Impostare come normale.

PutTime

Descrizione: L'ora in cui è stato inserito il messaggio.
Tipo di dati: MQCHAR8.
Valore: Impostare come normale.

ApplOriginData

Descrizione: Dati di applicazione relativi all'origine.
Tipo di dati: MQCHAR4.
Valore: Imposta come normale ..

Messaggio di traccia - instradamento MQEPH (intestazione PCF integrata)

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQEPH per un messaggio traceroute

La struttura MQEPH contiene una descrizione delle informazioni PCF che accompagnano i dati del messaggio di un messaggio traceroute e i dati del messaggio dell'applicazione che lo seguono. Una struttura MQEPH viene utilizzata solo se ulteriori dati del messaggio utente seguono il gruppo PCF TraceRoute .

Per un messaggio di tracce - route, la struttura MQEPH contiene i seguenti valori:

StrucId

Descrizione:	Identificatore struttura.
Tipo di dati:	MQCHAR4.
Valore:	ID_STRUC_MQEPH.

Version

Descrizione:	Numero di versione della struttura.
Tipo di dati:	MQLONG.
Valori:	MQEPH_VERSION_1.

StrucLength

Descrizione:	Lunghezza della struttura.
Tipo di dati:	MQLONG.
Valore:	Lunghezza totale della struttura che include le strutture del parametro PCF che la seguono.

Encoding

Descrizione:	Codifica numerica dei dati del messaggio che seguono l'ultima struttura del parametro PCF.
Tipo di dati:	MQLONG.
Valore:	La codifica dei dati del messaggio.

CodedCharSetId

Descrizione:	L'identificativo della serie di caratteri dei dati del messaggio che seguono l'ultima struttura del parametro PCF.
Tipo di dati:	MQLONG.
Valore:	La serie di caratteri dei dati del messaggio.

Format

Descrizione:	Nome formato dei dati del messaggio che seguono l'ultima struttura di parametro PCF.
Tipo di dati:	MQCHAR8.
Valore:	Il nome formato dei dati del messaggio.

Flags

Descrizione:	Indicatori che specificano gli attributi della struttura o ne controllano l'elaborazione.
Tipo di dati:	MQLONG.
Valore:	MQEPH_NONE Nessun indicatore specificato. MQEPH_CCSDID_EMBEDDED Specifica che la serie di caratteri dei parametri contenenti i dati carattere viene specificata singolarmente all'interno del campo <i>CodedCharSetId</i> in ogni struttura.

PCFHeader

Descrizione:	Intestazione formato comando programmabile
Tipo di dati:	MQCFH.
Valore:	Consultare “Messaggio di instradamento traccia MQCFH (intestazione PCF)” a pagina 134.

Messaggio di instradamento traccia MQCFH (intestazione PCF)

Utilizzare questa pagina per visualizzare i valori PCF contenuti nella struttura MQCFH per un messaggio di traceroute

Per un messaggio di traceroute, la struttura MQCFH contiene i seguenti valori:

Type

Descrizione:	Tipo di struttura che identifica il contenuto del messaggio.
Tipo di dati:	MQLONG.
Valore:	MQCF_TRACE_ROUTE Il messaggio è un messaggio di indirizzamento traccia.

StrucLength

Descrizione:	Lunghezza della struttura.
Tipo di dati:	MQLONG.
Valore:	LUNGHEZZA_STRUTTURA_MQCFH_STRUCT Lunghezza in byte della struttura MQCFH.

Version

Descrizione:	Numero di versione della struttura.
Tipo di dati:	MQLONG.
Valori:	MQCFH_VERSION_3

Command

Descrizione:	Identificativo comando. Identifica la categoria del messaggio.
Tipo di dati:	MQLONG.
Valori:	MQCMD_TRACE_ROUTE Messaggio di indirizzamento traccia.

MsgSeqNumber

Descrizione:	Numero di sequenza del messaggio. Questo è il numero di sequenza del messaggio all'interno di un gruppo di messaggi correlati.
Tipo di dati:	MQLONG.
Valori:	1.

Control

Descrizione:	Opzioni di controllo.
Tipo di dati:	MQLONG.
Valori:	MQCFC_LAST.

CompCode

Descrizione:	Codice di completamento.
Tipo di dati:	MQLONG.
Valori:	MQCC_OK.

Reason

Descrizione:	Codice di errore che qualifica il codice di completamento.
Tipo di dati:	MQLONG.
Valori:	MQRC_NONE.

ParameterCount

Descrizione:	Conteggio delle strutture di parametri. Questo è il numero di strutture di parametri che seguono la struttura MQCFH. Una struttura di gruppi (MQCFGR) e le relative strutture di parametri incluse vengono conteggiate come una sola struttura.
Tipo di dati:	MQLONG.
Valori:	1 o superiore.

Traccia - dati del messaggio di instradamento

Utilizzare questa pagina per visualizzare i parametri che costituiscono la parte del gruppo PCF *TraceRoute* dei dati del messaggio di instradamento traccia

Il contenuto dei dati del messaggio di tracerouti dipende dal parametro *Accumulato* del gruppo PCF *TraceRoute* . I dati del messaggio di traccia - instradamento sono costituiti dal gruppo PCF *TraceRoute* e da zero o più gruppi PCF *Attività* . Il gruppo PCF *TraceRoute* è descritto in dettaglio in questo argomento. Fare riferimento alle relative informazioni per dettagli sul gruppo PCF *Attività* .

I dati del messaggio di traccia - instradamento contengono i parametri seguenti:

TraceRoute

Descrizione:	Parametri raggruppati che specificano gli attributi del messaggio trace - route. Per un messaggio di traceroute, alcuni di questi parametri possono essere modificati per controllare come viene elaborato.
Identificativo:	MQGACF_TRACE_ROUTE.
Tipo di dati:	MQCFGR.
Contenuto nel gruppo PCF:	Nessuna.

Parametri nel gruppo:	<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

Descrizione:	Il livello di dettagli che verrà registrato per l'attività.
Identificativo:	MQIACF_ROUTE_DETAIL.
Tipo di dati:	MQCFIN.
Contenuto nel gruppo PCF:	<i>TraceRoute</i> .
Valori:	<p>MQROUTE_DETAIL_LOW Le attività eseguite dall'applicazione scritta dall'utente vengono registrate.</p> <p>MQROUTE_DETAIL_MEDIO Le attività specificate in MQROUTE_DETAIL_LOW vengono registrate. Inoltre, vengono registrate le attività eseguite dagli MCA.</p> <p>MQROUTE_DETAIL_HIGH Vengono registrate le attività specificate in MQROUTE_DETAIL_LOW e MQROUTE_DETAIL_MEDIUM. Gli MCA non registrano ulteriori informazioni sull'attività a questo livello di dettagli. Questa opzione è disponibile solo per le applicazioni scritte dall'utente che devono registrare ulteriori informazioni sull'attività.</p>

RecordedActivities

Descrizione:	Il numero di attività che il messaggio di traccia - instradamento ha causato, dove sono state registrate le informazioni.
Identificativo:	MQIACF_RECORDED_ACTIVITIES.
Tipo di dati:	MQCFIN.
Contenuto nel gruppo PCF:	<i>TraceRoute</i> .

UnrecordedActivities

Descrizione:	Il numero di attività che il messaggio di traccia - instradamento ha causato, dove le informazioni non sono state registrate.
Identificativo:	MQIACF_UNRECORDED_ACTIVITIES.
Tipo di dati:	MQCFIN.
Contenuto nel gruppo PCF:	<i>TraceRoute</i> .

DiscontinuityCount

Descrizione:	Il numero di volte in cui un messaggio di traceroute è stato ricevuto da un gestore code che non supporta la messaggistica traceroute.
--------------	--

Identificativo: MQIACF_DISCONTINUITY_COUNT.
Tipo di dati: MQCFIN.
Contenuto nel gruppo PCF: *TraceRoute*.

MaxActivities

Descrizione: Il numero massimo di attività in cui il messaggio di traccia - instradamento può essere coinvolto prima che smetta di essere elaborato.

Identificativo: MQIACF_MAX_ACTIVITIES.
Tipo di dati: MQCFIN.
Contenuto nel gruppo PCF: *TraceRoute*.

Valore: **Un numero intero positivo**
Il numero massimo di attività.
MQROUTE_UNLIMITED_ACTIVITIES
Un numero illimitato di attività.

Accumulate

Descrizione: Specifica se le informazioni sull'attività vengono accumulate all'interno del messaggio di instradamento della traccia e se viene generato un messaggio di risposta contenente le informazioni sull'attività accumulate prima che il messaggio di instradamento della traccia venga eliminato o venga inserito in una coda di non trasmissione.

Identificativo: MQIACF_ROUTE_ACCUMULO.
Tipo di dati: MQCFIN.
Contenuto nel gruppo PCF: *TraceRoute*.

Valore: **MQROUTE_ACCUMULATE_NONE**
Le informazioni sull'attività non vengono accumulate nei dati del messaggio di traccia - instradamento.
MQROUTE_ACCUMULATE_IN_MSG
Le informazioni sull'attività vengono accumulate nei dati del messaggio di traccia - instradamento.
MQROUTE_ACCUMULATE_E_REPLY
Le informazioni sull'attività vengono raccolte nei dati del messaggio di instradamento traccia e verrà generato un messaggio di risposta di instradamento traccia.

Forward

Descrizione: Specifica i gestori code a cui può essere inoltrato il messaggio di tracce - route. Quando si determina se inoltrare un messaggio a un gestore code remoto, i gestori code utilizzano l'algoritmo descritto in [Inoltro](#).

Identificativo: MQIACF_ROUTE_FORWARDING.
Tipo di dati: MQCFIN.
Contenuto nel gruppo PCF: *TraceRoute*.

Valore: **MQROUTE_FORWARD_IF_SUPPORTED**
Il messaggio trace - route viene inoltrato solo ai gestori code che onoreranno il valore del parametro *Deliver* del gruppo *TraceRoute* .

MQROUTE_FORWARD_ALL
Il messaggio di trace - route viene inoltrato a qualsiasi gestore code, indipendentemente dal fatto che il valore del parametro *Deliver* verrà rispettato.

Deliver

Descrizione: Specifica l'azione da intraprendere se il messaggio di trace - route arriva alla coda di destinazione con esito positivo.

Identificativo: MQIACF_ROUTE_DELIVERY.

Tipo di dati: MQCFIN.

Contenuto nel gruppo PCF: *TraceRoute*.

Valore: **MQROUTE_DELIVER_SÌ**
All'arrivo, il messaggio di trace - route viene inserito nella coda di destinazione. Qualsiasi applicazione che esegue un richiamo distruttivo sulla coda di destinazione può ricevere il messaggio di indirizzamento traccia.

MQROUTE_DELIVER_NO
All'arrivo, il messaggio trace - route viene eliminato.

Riferimento messaggio di risposta tracerouto

Utilizzare questa pagina per ottenere una panoramica del formato del messaggio di risposta trace - route. I dati del messaggio di risposta di trace - route sono un duplicato dei dati del messaggio di trace - route per cui sono stati creati

Formato del messaggio di risposta di tracerouto

I messaggi di risposta di traccia - instradamento sono messaggi IBM MQ standard che contengono un descrittore di messaggi e dati di messaggi. I dati del messaggio contengono informazioni sulle attività eseguite su un messaggio di trace - route come è stato instradato attraverso una rete di gestori code.

I messaggi di risposta di traccia - instradamento contengono le seguenti informazioni:

Un descrittore di messaggi

Una struttura MQMD

Dati messaggio

Un'intestazione PCF (MQCFH) e i dati del messaggio di risposta trace - route

I dati del messaggio di risposta di traccia - instradamento sono costituiti da uno o più gruppi PCF *Attività* .

Quando un messaggio di trace - route raggiunge la coda di destinazione, è possibile generare un messaggio di risposta di trace - route che contiene una copia delle informazioni sull'attività dal messaggio di trace - route. Il messaggio di risposta di instradamento della traccia verrà consegnato a una coda di risposta o a una coda di sistema.

Tabella 21 a pagina 139 mostra la struttura di un messaggio di risposta di tracerouto, inclusi i parametri che vengono restituiti solo in alcune condizioni.

Tabella 21. Formato del messaggio di risposta di traceroute

Struttura MQMD	Struttura MQCFH intestazione PCF	Traccia - dati del messaggio di risposta instradamento
Identificativo struttura Versione della struttura Opzioni di documentazione Tipo messaggio Ora di scadenza Feedback Codifica ID set di caratteri codificato Formato del messaggio Priorità Persistenza ID messaggio Identificativo di correlazione Conteggio backout Coda di risposta Gestore code di risposta Identificativo utente Token account Dati identità applicazione Tipo di applicazione Nome applicazione Data Put Ora Put Dati origine applicazione ID gruppo Numero sequenza messaggi Offset Indicatori di messaggio Lunghezza originale	Intestazione PCF (MQCFH) Tipo di struttura Lunghezza struttura Versione della struttura Identificativo comando Numero sequenza messaggi Opzioni di controllo Codice di completamento Codice di errore Conteggio parametro	Attività Nome applicazione attività Tipo di applicazione attività Descrizione attività Operazione Tipo di operazione Data operazione Ora operazione Messaggio Lunghezza messaggio MQMD EmbeddedMQMD Nome del gestore code Nome gruppo di condivisione code Nome coda ^{1 2 3} Nome coda risolta ^{1 3} Nome coda remota ³ Gestore code remoto - nome ^{2 3 4 5} Feedback ² Nome canale ^{4 5} Tipo di canale ^{5 5} Nome coda di trasmissione ⁵ TraceRoute Dettaglio Attività registrate Attività non registrate Conteggio discontinuità Attività massime Accumula Distribuzione

Nota:

1. Restituito per le operazioni Get e Browse.
2. Restituito per operazioni di eliminazione.
3. Restituito per le operazioni di inserimento, di inserimento risposta e di inserimento prospetto.
4. Restituito per operazioni di ricezione.
5. Restituito per le operazioni di invio.

MQMD (message descriptor) del messaggio di risposta trace - route

Utilizzare questa pagina per visualizzare i valori contenuti nella struttura MQMD per un messaggio di risposta traceroute

Per un messaggio di risposta di trace - route, la struttura MQMD contiene i parametri descritti in [Descrittore del messaggio di report attività](#). Alcuni dei valori di parametro in un descrittore di messaggi di risposta di traccia - instradamento sono diversi da quelli in un descrittore di messaggi di report di attività, come segue:

MsgType

Descrizione: Tipo di messaggio.
Tipo di dati: MQLONG.
Valore: **MQMT_REPLY**

Feedback

Descrizione: Feedback o codice di errore.
Tipo di dati: MQLONG.
Valore: **MQFB_NONE**

Encoding

Descrizione: Codifica numerica dei dati del messaggio.
Tipo di dati: MQLONG.
Valore: Copiato dal descrittore del messaggio di trace - route.

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio.
Tipo di dati: MQLONG.
Valore: Copiato dal descrittore del messaggio di trace - route.

Format

Descrizione: Nome formato dei dati di messaggio
Tipo di dati: MQCHAR8.
Valore: **MMQFMT_ADMIN**
Messaggio di amministrazione.

Messaggio di risposta traccia - instradamento MQCFH (intestazione PCF)

Utilizzare questa pagina per visualizzare i valori PCF contenuti nella struttura MQCFH per un messaggio di risposta traceroute

L'intestazione PCF (MQCFH) per un messaggio di risposta trace - route è uguale a quella per un messaggio trace - route.

Traccia - dati del messaggio di risposta instradamento

I dati del messaggio di risposta di trace - route sono un duplicato dei dati del messaggio di trace - route per cui sono stati creati

I dati del messaggio di risposta di trace - route contengono uno o più gruppi *Attività* . I parametri sono descritti in [“Dati del messaggio del report attività” a pagina 110.](#)

Messaggi di account e statistiche


I gestori code generano messaggi di account e statistiche per registrare informazioni sulle operazioni MQI eseguite dalle applicazioni IBM MQ o per registrare informazioni sulle attività che si verificano in un sistema IBM MQ .

Messaggi di account

I messaggi di account vengono utilizzati per registrare informazioni sulle operazioni MQI eseguite dalle applicazioni IBM MQ , consultare [“Messaggi di account” a pagina 141.](#)

Messaggi statistiche

I messaggi statistici vengono utilizzati per registrare le informazioni sulle attività che si verificano in un sistema IBM MQ, consultare [“Messaggi statistiche”](#) a pagina 144. Alcune attività registrate nei messaggi delle statistiche sono relative alle operazioni interne del gestore code.

 I messaggi di account e i messaggi di statistiche come qui descritti non sono disponibili su IBM MQ for z/OS, ma la funzione equivalente è disponibile tramite SMF (System Management Facility).

I messaggi di account e statistiche vengono consegnati a una delle due code di sistema. Le applicazioni utente possono recuperare i messaggi da queste code di sistema e utilizzare le informazioni registrate per vari scopi:

- Account per l'utilizzo delle risorse dell'applicazione.
- Registrare l'attività dell'applicazione.
- Pianificazione della capacità.
- Rileva problemi nella rete del gestore code.
- Assistenza nella determinazione delle cause dei problemi nella rete del gestore code.
- Migliora l'efficienza della tua rete di gestori code.
- Familiarizzare con l'esecuzione della rete del gestore code.
- Confermare che la rete del gestore code sia in esecuzione correttamente.

Concetti correlati

[“Using System Management Facility”](#) a pagina 329

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.

Messaggi di account

I messaggi di account registrano le informazioni sulle operazioni MQI eseguite dalle applicazioni IBM MQ. Un messaggio di account è un messaggio PCF che contiene un numero di strutture PCF.

Quando un'applicazione si disconnette da un gestore code, viene generato un messaggio di account che viene consegnato alla coda di account di sistema (SYSTEM.ADMIN.ACCOUNTING.QUEUE). Per le applicazioni IBM MQ di lunga durata, i messaggi di account intermedi vengono generati come segue:

- Quando il tempo trascorso dal momento in cui è stata stabilita la connessione supera l'intervallo configurato.
- Quando il tempo trascorso dall'ultimo messaggio di account intermedio supera l'intervallo configurato.

I messaggi di account sono nelle categorie seguenti:

Messaggi di account MQI

I messaggi di account MQI contengono informazioni sul numero di chiamate MQI effettuate utilizzando una connessione a un gestore code.

Messaggi di account coda

I messaggi di account della coda contengono informazioni relative al numero di chiamate MQI effettuate utilizzando le connessioni a un gestore code, raggruppate per coda.

Ogni messaggio di account coda può contenere fino a 100 record, con ogni record relativo ad un'attività eseguita dall'applicazione rispetto ad una coda specifica.

I messaggi di account vengono registrati solo per code locali. Se un'applicazione effettua una chiamata MQI rispetto ad una coda alias, i dati di account vengono registrati rispetto alla coda di base e, per una coda remota, i dati di account vengono registrati rispetto alla coda di trasmissione.

Nota: Poiché le informazioni si riferiscono alle operazioni di MQI eseguite dalle applicazioni IBM MQ, le informazioni non includono le operazioni relative a una coda di flusso, a meno che l'operazione non sia direttamente su tale coda dall'applicazione.

Riferimenti correlati

“Dati del messaggio di account MQI” a pagina 160

Utilizzare questa pagina per visualizzare la struttura di un messaggio di account MQI

“Dati del messaggio di account coda” a pagina 172

Utilizzare questa pagina per visualizzare la struttura di un messaggio di account coda

Formato del messaggio di account

I messaggi di account comprendono una serie di campi PCF costituiti da un descrittore di messaggi e dati di messaggi.

Descrittore messaggio

- Un messaggio di account MQMD (descrittore del messaggio)

Dati del messaggio di account

- Un messaggio di account MQCFH (intestazione PCF)
- Dati del messaggio di account sempre restituiti
- Dati del messaggio di account restituiti se disponibili

Il messaggio di account MQCFH (intestazione PCF) contiene informazioni sull'applicazione e l'intervallo per cui sono stati registrati i dati di account.

I dati del messaggio di account comprendono parametri PCF che memorizzano le informazioni di account. Il contenuto dei messaggi di account dipende dalla categoria di messaggi come segue:

Messaggio di account MQI

I dati del messaggio di account MQI sono costituiti da un numero di parametri PCF, ma non da gruppi PCF.

Messaggio di account coda

I dati del messaggio di account della coda sono costituiti da un numero di parametri PCF e sono compresi nell'intervallo tra 1 e 100 *QAccountingData* gruppi PCF.

Esiste un gruppo PCF *QAccountingData* per ogni coda per cui sono stati raccolti dati di account. Se un'applicazione accede a più di 100 code, vengono generati più messaggi di account. Ogni messaggio ha *SeqNumber* in MQCFH (intestazione PCF) aggiornato di conseguenza e l'ultimo messaggio nella sequenza ha il parametro *Control* in MQCFH specificato come MQCFC_LAST.

Raccolta di informazioni contabili

Utilizzare gli attributi di coda e gestore code per controllare la raccolta di informazioni di account. È anche possibile utilizzare opzioni MQCONNX per controllare la raccolta a livello di connessione.

Controllo della raccolta delle informazioni di account MQI

Utilizzare l'attributo gestore code ACCTMQI per controllare la raccolta di informazioni di account MQI.

Per modificare il valore di questo attributo, utilizzare il comando MQSC, ALTER QMGR specificare il parametro ACCTMQI. I messaggi di account vengono creati solo per connessioni che iniziano dopo l'abilitazione dell'account. Il parametro **ACCTMQI** può avere i seguenti valori:

Attivo

Le informazioni di account MQI vengono raccolte per ogni connessione al gestore code.

Disattivo

Le informazioni di account MQI non vengono raccolte. Questo è il valore predefinito.

Ad esempio, per abilitare la raccolta di informazioni di account MQI, utilizzare il seguente comando MQSC:

```
ALTER QMGR ACCTMQI(ON)
```

Informazioni sull'account della coda

Utilizzare l'attributo della coda ACCTQ e l'attributo del gestore code ACCTQ per controllare la raccolta di informazioni sull'account coda.

Per modificare il valore dell'attributo della coda, utilizzare il comando MQSC ALTER QLOCAL e specificare il parametro ACCTQ. I messaggi di account vengono creati solo per connessioni che iniziano dopo l'abilitazione dell'account. Tenere presente che le modifiche a questo valore sono effettive solo per le connessioni al gestore code che si verificano dopo la modifica all'attributo.

L'attributo della coda ACCTQ può avere i seguenti valori:

Attivo

Le informazioni di account coda per questa coda vengono raccolte per ogni connessione al gestore code che apre la coda.

Disattivo

Le informazioni di account coda per questa coda non sono raccolte.

QMGR

La raccolta delle informazioni di account coda per questa coda è controllata in base al valore dell'attributo del gestore code ACCTQ. Questo è il valore predefinito.

Per modificare il valore dell'attributo del gestore code, utilizzare il comando MQSC ALTER QMGR e specificare il parametro ACCTQ. L'attributo del gestore code ACCTQ può avere i seguenti valori:

Attivo

Le informazioni sull'account della coda vengono raccolte per le code per cui l'attributo della coda ACCTQ è impostato come QMGR.

Disattivo

Le informazioni sull'account della coda non vengono raccolte per le code che hanno l'attributo della coda ACCTQ impostato come QMGR. Questo è il valore predefinito.

Nessuna

La raccolta di informazioni sull'account della coda è disabilitata per tutte le code, indipendentemente dall'attributo coda ACCTQ.

Se l'attributo del gestore code, ACCTQ, è impostato su NONE, la raccolta delle informazioni sull'account della coda è disabilitata per tutte le code, indipendentemente dall'attributo della coda ACCTQ.

Ad esempio, per abilitare la raccolta di informazioni di account per la coda, Q1, utilizzare il seguente comando MQSC:

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

Per abilitare la raccolta di informazioni di account per tutte le code che specificano l'attributo coda ACCTQ come QMGR, utilizzare il seguente comando MQSC:

```
ALTER QMGR ACCTQ(ON)
```

Opzioni MQCONNX per il controllo della raccolta delle informazioni di account

Utilizzare il parametro **ConnectOpts** sulla chiamata MQCONNX per modificare la raccolta di informazioni di account della coda e MQI a livello di connessione sovrascrivendo i valori effettivi degli attributi del gestore code ACCTMQI e ACCTQ

Il parametro **ConnectOpts** può avere i seguenti valori:

MQCNO_ACCOUNTING_MQI_ENABLED

Se il valore dell'attributo del gestore code ACCTMQI è specificato come OFF, l'account MQI è abilitato per questa connessione. Questo è l'equivalente dell'attributo del gestore code ACCTMQI specificato come ON.

Se il valore dell'attributo gestore code ACCTMQI non è specificato come OFF, questo attributo non ha effetto.

MQCNO_ACCOUNTING_MQI_DISABLED

Se il valore dell'attributo del gestore code ACCTMQI è specificato come ON, l'account MQI è disabilitato per questa connessione. Equivale all'attributo del gestore code ACCTMQI specificato come OFF.

Se il valore dell'attributo del gestore code ACCTMQI non è specificato come ON, questo attributo non ha alcun effetto.

MQCNO_ACCOUNTING_Q_ENABLED

Se il valore dell'attributo del gestore code ACCTQ è specificato come OFF, l'account della coda è abilitato per questa connessione. Tutte le code con ACCTQ specificato come QMGR sono abilitate per l'account della coda. Equivale all'attributo del gestore code ACCTQ specificato come ON.

Se il valore dell'attributo del gestore code ACCTQ non è specificato come OFF, questo attributo non ha alcun effetto.

MQCNO_ACCOUNTING_Q_DISABLED

Se il valore dell'attributo del gestore code ACCTQ è specificato come ON, l'account della coda è disabilitato per questa connessione. Equivale all'attributo del gestore code ACCTQ specificato come OFF.

Se il valore dell'attributo del gestore code ACCTQ non è specificato come ON, questo attributo non ha alcun effetto.

Queste sovrascritture sono disabilitate per impostazione predefinita. Per abilitarli, impostare l'attributo gestore code ACCTCONO su ENABLED. Per abilitare le sovrascritture di account per le singole connessioni, utilizzare il seguente comando MQSC:

```
ALTER QMGR ACCTCONO(ENABLED)
```

Generazione del messaggio di account quando le applicazioni si disconnettono

I messaggi di account vengono generati quando un'applicazione si disconnette dal gestore code. I messaggi di account intermedi vengono scritti anche per applicazioni IBM MQ di lunga durata.

I messaggi di account vengono generati in uno dei modi seguenti quando un'applicazione si disconnette:

- L'applicazione emette una chiamata MQDISC
- Il gestore code riconosce che l'applicazione è stata terminata

I messaggi di account intermedi vengono scritti per le applicazioni IBM MQ di lunga durata quando l'intervallo da quando è stata stabilita la connessione o dall'ultimo messaggio di account intermedio scritto supera l'intervallo configurato. L'attributo gestore code, ACCTINT, specifica il tempo, in secondi, dopo il quale i messaggi di account intermedi possono essere scritti automaticamente. I messaggi di account vengono creati solo quando l'applicazione interagisce con il gestore code, quindi le applicazioni che rimangono connesse al gestore code per lunghi periodi senza eseguire richieste MQI non generano messaggi di account fino all'esecuzione della prima richiesta MQI dopo il completamento dell'intervallo di account.

L'intervallo di account predefinito è 1800 secondi (30 minuti). Ad esempio, per modificare l'intervallo di account in 900 secondi (15 minuti), utilizzare il seguente comando MQSC:

```
ALTER QMGR ACCTINT(900)
```

Messaggi statistiche

I messaggi delle statistiche registrano informazioni sulle attività che si verificano in un sistema IBM MQ. Un messaggio di statistiche è un messaggio PCF che contiene un numero di strutture PCF.

I messaggi statistici vengono consegnati alla coda di sistema (SYSTEM.ADMIN.STATISTICS.QUEUE) ad intervalli configurati, ogni volta che vi è qualche attività.

I messaggi di statistiche sono nelle seguenti categorie:

Messaggi di statistiche MQI

I messaggi delle statistiche MQI contengono informazioni relative al numero di chiamate MQI effettuate durante un intervallo configurato. Ad esempio, le informazioni possono includere il numero di chiamate MQI emesse da un gestore code.

Messaggi statistiche coda

I messaggi di statistiche coda contengono informazioni relative all'attività di una coda durante un intervallo configurato. Le informazioni includono il numero di messaggi immessi e richiamati dalla coda e il numero totale di byte elaborati da una coda.

Ogni messaggio di statistica della coda può contenere fino a 100 record, con ogni record relativo all'attività per coda per cui sono state raccolte le statistiche.

I messaggi statistici vengono registrati solo per code locali. Se un'applicazione effettua una chiamata MQI su una coda alias, i dati delle statistiche vengono registrati rispetto alla coda di base e, per una coda remota, i dati delle statistiche vengono registrati rispetto alla coda di trasmissione.

Messaggi statistiche canale

I messaggi delle statistiche del canale contengono informazioni relative all'attività di un canale durante un intervallo configurato. Ad esempio, le informazioni possono essere il numero di messaggi trasferiti dal canale o il numero di byte trasferiti dal canale.

Ogni messaggio di statistiche del canale contiene fino a 100 record, con ciascun record relativo all'attività per canale per cui sono state raccolte le statistiche.

Nota: Poiché le informazioni sono relative alle attività che si verificano in un sistema IBM MQ , le informazioni includono le operazioni relative alle code di flusso.

Riferimenti correlati

[“Controllo della raccolta di statistiche MQI” a pagina 146](#)

Utilizzare l'attributo gestore code STATMQI per controllare la raccolta di informazioni sulle statistiche MQI.

[“Controllo della raccolta delle statistiche della coda” a pagina 146](#)

Utilizzare l'attributo della coda STATQ e l'attributo del gestore code STATQ per controllare la raccolta di informazioni sulle statistiche della coda

[“Controllo della raccolta delle statistiche del canale” a pagina 147](#)

Utilizzare l'attributo del canale STATCHL per controllare la raccolta di informazioni sulle statistiche del canale. È anche possibile impostare gli attributi del gestore code per controllare la raccolta di informazioni.

Formato dei messaggi di statistiche

I messaggi statistici comprendono una serie di campi PCF costituiti da un descrittore di messaggi e da dati di messaggi.

Descrittore messaggio

- Un messaggio statistico MQMD (descrittore del messaggio)

Dati del messaggio di account

- Un messaggio di statistiche MQCFH (intestazione PCF)
- Dati del messaggio delle statistiche sempre restituiti
- Dati del messaggio delle statistiche restituiti se disponibili

Il messaggio delle statistiche MQCFH (intestazione PCF) contiene informazioni sull'intervallo per cui sono stati registrati i dati delle statistiche.

I dati del messaggio delle statistiche comprendono parametri PCF che memorizzano le informazioni sulle statistiche. Il contenuto dei messaggi statistici dipende dalla categoria di messaggi come segue:

Messaggi statistiche MQI

I dati del messaggio delle statistiche MQI sono costituiti da un numero di parametri PCF, ma non da gruppi PCF.

Messaggio statistiche coda

I dati dei messaggi delle statistiche della coda sono costituiti da un numero di parametri PCF e nell'intervallo compreso tra 1 e 100 *QStatisticsData* gruppi PCF.

È presente un gruppo PCF *QStatisticsData* per ogni coda attiva nell'intervallo. Se nell'intervallo erano attive più di 100 code, vengono generati più messaggi di statistiche. Ogni messaggio ha *SeqNumber* in MQCFH (intestazione PCF) aggiornato di conseguenza e l'ultimo messaggio nella sequenza ha il parametro *Control* in MQCFH specificato come MQCFC_LAST.

Messaggio statistiche canale

I dati dei messaggi delle statistiche dei canali sono costituiti da un numero di parametri PCF e sono compresi tra 1 e 100 *ChlStatisticsData* gruppi PCF.

Esiste un gruppo PCF *ChlStatisticsData* per ogni canale attivo nell'intervallo. Se nell'intervallo erano attivi più di 100 canali, vengono generati più messaggi statistici. Ogni messaggio ha *SeqNumber* in MQCFH (intestazione PCF) aggiornato di conseguenza e l'ultimo messaggio nella sequenza ha il parametro *Control* in MQCFH specificato come MQCFC_LAST.

Raccolta di informazioni statistiche

Utilizzare gli attributi di coda, gestore code e canale per controllare la raccolta di informazioni statistiche

Controllo della raccolta di statistiche MQI

Utilizzare l'attributo gestore code STATMQI per controllare la raccolta di informazioni sulle statistiche MQI.

Per modificare il valore di questo attributo, utilizzare il comando MQSC ALTER QMGR e specificare il parametro **STATMQI**. I messaggi di statistiche vengono generati solo per le code aperte dopo l'abilitazione della raccolta delle statistiche. Il parametro **STATMQI** può avere i seguenti valori:

Attivo

Le informazioni statistiche MQI vengono raccolte per ogni connessione al gestore code.

Disattivo

Le informazioni statistiche MQI non vengono raccolte. Questo è il valore predefinito.

Ad esempio, per abilitare la raccolta di informazioni sulle statistiche MQI, utilizzare il seguente comando MQSC:

```
ALTER QMGR STATMQI(ON)
```

Controllo della raccolta delle statistiche della coda

Utilizzare l'attributo della coda STATQ e l'attributo del gestore code STATQ per controllare la raccolta di informazioni sulle statistiche della coda

È possibile abilitare o disabilitare la raccolta di informazioni sulle statistiche della coda per le singole code o per più code. Per controllare le code individuali, impostare l'attributo della coda STATQ. Abilitare o disabilitare la raccolta di informazioni sulle statistiche della coda a livello di gestore code utilizzando l'attributo STATQ del gestore code. Per tutte le code che hanno l'attributo della coda STATQ specificato con il valore QMGR, la raccolta di informazioni sulle statistiche della coda è controllata al livello del gestore code.

Le statistiche della coda vengono incrementate solo per le operazioni che utilizzano gli handle dell'oggetto MQI IBM MQ aperti dopo l'abilitazione della raccolta delle statistiche.

I messaggi delle statistiche della coda vengono generati solo per le code per le quali sono stati raccolti dati statistici nel periodo di tempo precedente.

La stessa coda può avere diverse operazioni di inserimento e operazioni di acquisizione attraverso diversi handle di oggetti. Alcuni handle di oggetti potrebbero essere stati aperti prima dell'abilitazione della raccolta di statistiche, ma altri sono stati aperti in seguito. Pertanto, è possibile per le statistiche della coda registrare l'attività di alcune operazioni di inserimento e di acquisizione, e non tutte.

Per assicurarsi che le statistiche della coda stiano registrando l'attività di tutte le applicazioni, è necessario chiudere e riaprire i nuovi handle di oggetto sulla coda o sulle code che si stanno monitorando. Il modo migliore per ottenere questo risultato è quello di terminare e riavviare tutte le applicazioni dopo aver abilitato la raccolta delle statistiche.

Per modificare il valore dell'attributo della coda STATQ, utilizzare il comando MQSC ALTER QLOCAL e specificare STATQ. L'attributo della coda STATQ può avere i seguenti valori:

Attivo

Le informazioni sulle statistiche della coda vengono raccolte per ogni connessione al gestore code che apre la coda.

Disattivo

Le informazioni sulle statistiche della coda per questa coda non vengono raccolte.

QMGR

La raccolta delle informazioni sulle statistiche della coda per questa coda è controllata in base al valore dell'attributo gestore code, STATQ. Questo è il valore predefinito.

Per modificare il valore dell'attributo gestore code STATQ, utilizzare il comando MQSC ALTER QMGR e specificare STATQ. L'attributo STATQ del gestore code può avere i seguenti valori:

Attivo

Le informazioni sulle statistiche della coda vengono raccolte per le code che hanno l'attributo della coda STATQ impostato come QMGR

Disattivo

Le informazioni sulle statistiche della coda non vengono raccolte per le code che hanno l'attributo della coda STATQ impostato come QMGR. Questo è il valore predefinito.

Nessuna

La raccolta delle informazioni statistiche della coda è disabilitata per tutte le code, indipendentemente dall'attributo della coda STATQ.

Se l'attributo STATQ del gestore code è impostato su NONE, la raccolta di informazioni sulle statistiche della coda è disabilitata per tutte le code, indipendentemente dall'attributo STATQ della coda.

Ad esempio, per abilitare la raccolta di informazioni sulle statistiche per la coda, Q1, utilizzare il seguente comando MQSC:

```
ALTER QLOCAL(Q1) STATQ(ON)
```

Per abilitare la raccolta di informazioni statistiche per tutte le code che specificano l'attributo della coda STATQ come QMGR, utilizzare il seguente comando MQSC:

```
ALTER QMGR STATQ(ON)
```

Controllo della raccolta delle statistiche del canale

Utilizzare l'attributo del canale STATCHL per controllare la raccolta di informazioni sulle statistiche del canale. È anche possibile impostare gli attributi del gestore code per controllare la raccolta di informazioni.

È possibile abilitare o disabilitare la raccolta di informazioni sulle statistiche del canale per singoli canali o per più canali. Per controllare i singoli canali, è necessario impostare l'attributo del canale STATCHL per abilitare o disabilitare la raccolta di informazioni sulle statistiche del canale. Per controllare insieme molti canali, abilitare o disabilitare la raccolta di informazioni sulle statistiche dei canali a livello di gestore code utilizzando l'attributo STATCHL del gestore code. Per tutti i canali con l'attributo del canale STATCHL specificato con il valore QMGR, la raccolta di informazioni sulle statistiche del canale è controllata a livello del gestore code.

I canali mittente del cluster definiti automaticamente non sono oggetti IBM MQ, quindi non hanno attributi allo stesso modo degli oggetti del canale. Per controllare automaticamente i canali mittenti del cluster definiti, utilizzare l'attributo STATACLS del gestore code. Questo attributo determina se i canali

mittenti del cluster definiti automaticamente all'interno di un gestore code sono abilitati o disabilitati per la raccolta di informazioni sulle statistiche del canale.

È possibile impostare la raccolta di informazioni sulle statistiche del canale su uno dei tre livelli di monitoraggio: basso, medio o alto. È possibile impostare il livello di monitoraggio a livello di oggetto o a livello di gestore code. La scelta del livello da utilizzare dipende dal sistema. La raccolta di dati statistici potrebbe richiedere alcune istruzioni relativamente costose dal punto di vista del calcolo, quindi per ridurre l'impatto della raccolta di informazioni statistiche del canale, le opzioni di monitoraggio medio e basso misurano un campione di dati a intervalli regolari piuttosto che raccogliere dati per tutto il tempo. [Tabella 22 a pagina 148](#) riepiloga i livelli disponibili con la raccolta di informazioni sulle statistiche del canale:

Livello	Descrizione	Utilizzo
Bassa	Misurare un piccolo campione di dati, ad intervalli regolari.	Per oggetti che elaborano un volume elevato di messaggi.
Medio	Misurare un campione di dati, ad intervalli regolari.	Per la maggior parte degli oggetti.
Elevato	Misurare tutti i dati, a intervalli regolari.	Per gli oggetti che elaborano solo pochi messaggi al secondo, su cui sono importanti le informazioni più recenti.

Per modificare il valore dell'attributo del canale STATCHL, utilizzare il comando MQSC ALTER CHANNEL e specificare il parametro STATCHL.

Per modificare il valore dell'attributo gestore code STATCHL, utilizzare il comando MQSC ALTER QMGR e specificare STATCHL.

Per modificare il valore dell'attributo del gestore code STATACLS, utilizzare il comando MQSC ALTER QMGR e specificare il parametro STATACLS.

L'attributo del canale, STATCHL, può avere i seguenti valori:

BASSO

Le informazioni sulle statistiche del canale vengono raccolte con un basso livello di dettaglio.

MEDIO

Le informazioni statistiche del canale vengono raccolte con un livello medio di dettaglio.

ALTO

Le informazioni sulle statistiche del canale vengono raccolte con un alto livello di dettaglio.

Disattivo

Le informazioni sulle statistiche del canale non vengono raccolte per questo canale.

QMGR

L'attributo di canale è impostato come QMGR. La raccolta di informazioni statistiche per questo canale è controllata dal valore dell'attributo gestore code, STATCHL.

Questo è il valore predefinito.



Su z/OS sistemi, abilitando questo parametro si attiva semplicemente la raccolta dei dati statistici, indipendentemente dal valore selezionato. La specifica di LOW, MEDIUM o HIGH non comporta alcuna differenza sui risultati. Questo parametro deve essere abilitato al fine di raccogliere i record di contabilità di canale.

L'attributo del gestore code, STATCHL, può avere i seguenti valori:

BASSO

Le informazioni sulle statistiche del canale vengono raccolte con un basso livello di dettagli, per tutti i canali che hanno l'attributo del canale STATCHL impostato come QMGR.

MEDIO

Le informazioni sulle statistiche del canale vengono raccolte con un medio livello di dettaglio, per tutti i canali che hanno l'attributo del canale STATCHL impostato come QMGR.

ALTO

Le informazioni sulle statistiche del canale vengono raccolte con un livello di dettaglio elevato, per tutti i canali che hanno l'attributo del canale STATCHL impostato come QMGR.


Disattivo

Le informazioni sulle statistiche del canale non vengono raccolte per tutti i canali che hanno l'attributo del canale STATCHL impostato come QMGR.

Questo è il valore predefinito.

Nessuna

La raccolta delle informazioni sulle statistiche del canale è disabilitata per tutti i canali, indipendentemente dall'attributo del canale STATCHL.

 Su z/OS sistemi, abilitando questo parametro si attiva semplicemente la raccolta dei dati statistici, indipendentemente dal valore selezionato. La specifica di LOW, MEDIUM o HIGH non comporta alcuna differenza sui risultati. Questo parametro deve essere abilitato al fine di raccogliere i record di contabilità di canale.

L'attributo gestore code, STATACLS, può avere i seguenti valori:

BASSO

Le informazioni statistiche vengono raccolte con un basso livello di dettaglio per canali mittenti del cluster definiti automaticamente.

MEDIO

Le informazioni statistiche vengono raccolte con un medio livello di dettaglio per i canali mittenti del cluster definiti automaticamente.

ALTO

Le informazioni sulle statistiche vengono raccolte con un alto livello di dettagli per i canali mittenti del cluster definiti automaticamente.


Disattivo

Le informazioni statistiche non vengono raccolte per i canali mittenti del cluster definiti automaticamente.

QMGR

La raccolta di informazioni statistiche per canali mittente del cluster definiti automaticamente è controllata dal valore dell'attributo del gestore code, STATCHL.

Questo è il valore predefinito.

 Su z/OS sistemi, abilitando questo parametro si attiva semplicemente la raccolta dei dati statistici, indipendentemente dal valore selezionato. La specifica di LOW, MEDIUM o HIGH non comporta alcuna differenza sui risultati. Questo parametro deve essere abilitato al fine di raccogliere i record di contabilità di canale.

Ad esempio, per abilitare la raccolta di informazioni statistiche, con un medio livello di dettagli, per il canale mittente QM1 . TO . QM2, utilizzare il seguente comando MQSC:

```
ALTER CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

Per abilitare la raccolta delle informazioni statistiche, ad un livello medio di dettagli, per tutti i canali che specificano l'attributo del canale STATCHL come QMGR, utilizzare il seguente comando MQSC:

```
ALTER QMGR STATCHL(MEDIUM)
```

Per abilitare la raccolta di informazioni statistiche, a un livello medio di dettagli, per tutti i canali mittenti del cluster definiti automaticamente, utilizzare il seguente comando MQSC:

```
ALTER QMGR STATACLS(MEDIUM)
```

Creazione di messaggi di statistiche

I messaggi di statistica vengono generati a intervalli configurati e quando un gestore code viene arrestato in modo controllato.

L'intervallo configurato è controllato dall'attributo gestore code STATINT, che specifica l'intervallo, in secondi, tra la creazione dei messaggi di statistiche. L'intervallo di statistiche predefinito è 1800 secondi (30 minuti). Per modificare l'intervallo delle statistiche, utilizzare il comando MQSC ALTER QMGR e specificare il parametro **STATINT**. Ad esempio, per modificare l'intervallo delle statistiche in 900 secondi (15 minuti), utilizzare il seguente comando MQSC:

```
ALTER QMGR STATINT(900)
```

Per scrivere i dati delle statistiche attualmente raccolti nella coda delle statistiche prima della scadenza dell'intervallo di raccolta delle statistiche, utilizzare il comando MQSC RESET QMGR TYPE (STATISTICS). L'immissione di questo comando fa sì che i dati delle statistiche raccolti vengano scritti nella coda delle statistiche e che inizi un nuovo intervallo di raccolta dati delle statistiche.

Visualizzazione delle informazioni su statistiche e account

Per utilizzare le informazioni registrate nei messaggi di account e statistiche, eseguire un'applicazione come il programma di esempio **amqsmn** per trasformare le informazioni registrate in un formato appropriato

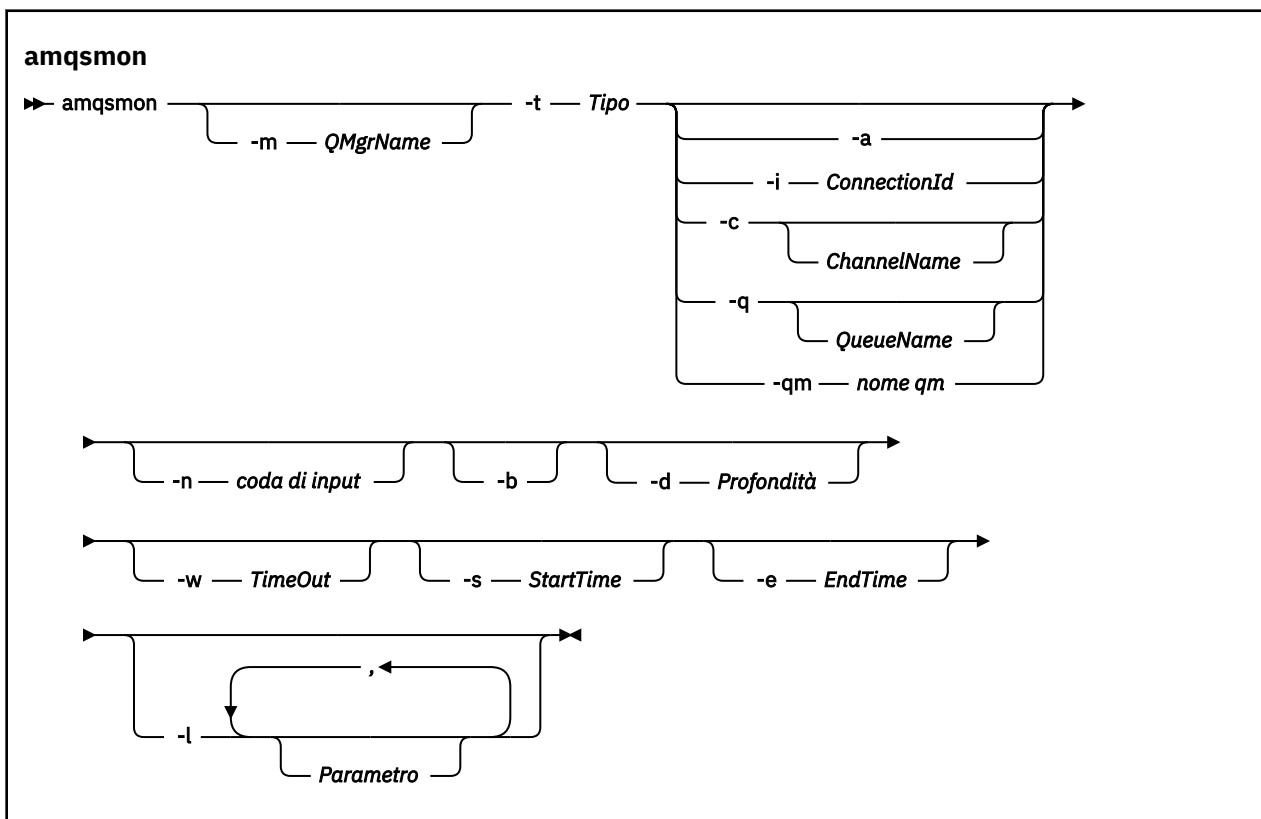
I messaggi di account e statistiche vengono scritti nelle code di account e statistiche del sistema. **amqsmn** è un programma di esempio fornito con IBM MQ che elabora i messaggi dalle code di account e statistiche e visualizza le informazioni sullo schermo in un formato leggibile.

Poiché **amqsmn** è un programma di esempio, è possibile utilizzare il codice di origine fornito come modello per scrivere la propria applicazione per elaborare i messaggi di statistiche o di account oppure modificare il codice di origine **amqsmn** per soddisfare i propri requisiti specifici.

amqsmn (Visualizzazione informazioni di controllo formattate)

Utilizzare il programma di esempio **amqsmn** per visualizzare in un formato leggibile le informazioni contenute nei messaggi di account e statistiche. Il programma **amqsmn** legge i messaggi di account dalla coda di account, SYSTEM.ADMIN.ACCOUNTING.QUEUE. e legge i messaggi statistici dalla coda delle statistiche, SYSTEM.ADMIN.STATISTICS.QUEUE.

Sintassi



Parametri obbligatori

-t *Tipo*

Il tipo di messaggi da elaborare. Specificare *Tipo* come uno dei seguenti:

contabilità

Vengono elaborati i record di account. I messaggi vengono letti dalla coda di sistema SYSTEM.ADMIN.ACCOUNTING.QUEUE a meno che non sia stato utilizzato il parametro -n per selezionare una coda particolare da cui leggere.

statistiche

I record delle statistiche vengono elaborati. I messaggi vengono letti dalla coda di sistema SYSTEM.ADMIN.STATISTICS.QUEUE **V 9.4.0** a meno che non sia stato utilizzato il parametro -n per selezionare una coda particolare da cui leggere.

Parametri facoltativi

-m *QMgrName*

Il nome del gestore code da cui devono essere elaborati i messaggi di statistiche o di account.

Se non si specifica questo parametro, viene utilizzato il gestore code predefinito.

-a

Elaborare i messaggi contenenti solo record MQI.

Visualizzare solo i record MQI. I messaggi non contenenti record MQI verranno sempre lasciati nella coda da cui sono stati letti.

-q *QueueName*

QueueName è un parametro facoltativo.

Se *QueueName* non viene fornito: Visualizza solo i record di statistiche coda e account coda.

Se viene fornito *QueueName* : Visualizza i record di statistiche coda e account della coda solo per la coda specificata da *QueueName* .

Se *-b* non è specificato, i messaggi di statistiche e di account da cui provengono i record vengono eliminati. Poiché i messaggi di account e statistiche possono contenere anche record provenienti da altre code, se *-b* non è specificato, i record non visualizzati possono essere eliminati.

-c ChannelName

ChannelName è un parametro facoltativo.

Se *ChannelName* non viene fornito: Visualizza solo i record delle statistiche di canale.

Se viene fornito *ChannelName* : Visualizza i record delle statistiche del canale solo per il canale specificato da *ChannelName* .

Se *-b* non viene specificato, i messaggi di statistiche da cui provengono i record vengono eliminati. Poiché i messaggi statistici possono contenere anche record provenienti da altri canali, se *-b* non è specificato, i record non visualizzati possono essere eliminati.

Questo parametro è disponibile solo quando vengono visualizzati i messaggi delle statistiche (*-t statistics*).

V 9.4.0 -qm nome qm

Questo parametro è facoltativo, ma se lo si specifica è necessario fornire un nome gestore code

Specifica che solo i messaggi di account o di statistiche prodotti dal gestore code fornito devono essere letti dalla coda. In genere, viene utilizzato solo se i messaggi provenienti da diversi gestori code sono stati inoltrati a un gestore code centrale.

-i ConnectionId

Visualizza i record relativi all'identificativo di connessione specificato solo da *ConnectionId* .

Questo parametro è disponibile solo quando si visualizzano i messaggi di account (*-t accounting*).

Se *-b* non viene specificato, i messaggi di statistiche da cui provengono i record vengono eliminati. Poiché i messaggi statistici possono contenere anche record provenienti da altri canali, se *-b* non è specificato, i record non visualizzati possono essere eliminati.

V 9.4.0 -n coda di input

La coda da cui leggere i messaggi di account o statistiche. Se non viene specificato, viene utilizzata la coda predefinita per il *-t < tipo >* scelto.

-b

Sfoggia messaggi.

I messaggi vengono richiamati in modo non distruttivo.

-d Profondità

Il numero massimo di messaggi che è possibile elaborare.

Se non si specifica questo parametro, è possibile elaborare un numero illimitato di messaggi.

-w TimeOut

Il numero massimo di secondi di attesa per la disponibilità di un messaggio.

Se non si specifica questo parametro, amqsmn terminerà una volta che non ci sono più messaggi da elaborare.

-s StartTime

Elabora i messaggi immessi solo dopo *StartTime* specificato.

StartTime è specificato nel formato yyyy-mm-dd hh.mm.ss. Se una data viene specificata senza un'ora, l'ora verrà impostata sul valore predefinito 00.00.00 nella data specificata. Gli orari sono GMT.

Per l'effetto di non specificare questo parametro, consultare la [Nota 1](#).

-e EndTime

Elabora i messaggi immessi prima del *EndTime* specificato.

EndTime è specificato nel formato yyyy-mm-dd hh.mm.ss. Se una data viene specificata senza un'ora, l'ora verrà impostata sul valore predefinito 23.59.59 nella data specificata. Gli orari sono GMT.

Per l'effetto di non specificare questo parametro, consultare la [Nota 1](#).

-l Parametro

Visualizzare solo i campi selezionati dai record elaborati. *Parametro* è un elenco separato da virgole di valori interi, con ogni valore intero associato alla costante numerica di un campo, vedere [amqsmn example 5](#).

Se non si specifica questo parametro, vengono visualizzati tutti i campi disponibili.

Nota:

1. Se non si specifica *-s StartTime* o *-e EndTime*, i messaggi che possono essere elaborati non vengono limitati dall'ora di inserimento.

Esempi di amqsmn (Visualizzazione informazioni di monitoraggio formattate)

Utilizzare questa pagina per visualizzare gli esempi di esecuzione del programma di esempio amqsmn (Visualizzazione delle informazioni di monitoraggio formattate)

1. Consultare [“Dati messaggio statistiche coda”](#) a pagina 194 per una spiegazione degli attributi.

Il seguente comando visualizza tutti i messaggi di statistiche MQI dal gestore code saturn.queue.manager:

```
amqsmn -m saturn.queue.manager -t statistics -a
```

L'output di questo comando è il seguente:

```
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. Il seguente comando visualizza tutti i messaggi di statistiche della coda per la coda LOCALQ sul gestore code saturn.queue.manager:

```
amqsmon -m saturn.queue.manager -t statistics -q LOCALQ
```

L'output di questo comando è il seguente:

```
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. Il seguente comando visualizza tutti i messaggi statistici registrati dalle 15:30 del 30 aprile 2005 dal gestore code saturn.queue.manager.

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

L'output di questo comando è il seguente:

```
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]
...
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. Consultare [“Dati del messaggio di account coda”](#) a pagina 172 per una spiegazione degli attributi.

Il seguente comando visualizza tutti i messaggi di account registrati il 30 aprile 2005 dal gestore code saturn.queue.manager:

```
amqsmom -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

L'output di questo comando è il seguente:

```

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, 0]
OpenFailCount: [0, 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, 0]
CloseFailCount: [0, 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, 0]
InqFailCount: [0, 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, 0]
SetFailCount: [0, 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. Il seguente comando sfoglia la coda di account e visualizza il nome dell'applicazione e l'identificativo di connessione di tutte le applicazioni per cui sono disponibili le informazioni di account MQI:

Il **Identifier** per **ApplicationName** è MQCACF_APPL_NAME e per **ConnectionId** è MQBACF_CONNECTION_ID

2. Ricercare gli ID trovati nel passo “1” a pagina 156.

Andare alla sezione Costanti e scorrere verso il basso fino a quando non si trova l'elenco MQCACF_* (Tipo di parametro carattere formato comando) . Individuare MQCACF_APPL_NAME e visualizzare il valore 3024.

Allo stesso modo, trovare l'elenco MQBACF_* (Tipi di parametro byte in formato comando). Individuare MQBACF_CONNECTION_ID e visualizzare il valore 7006.

Ricerca dell'associazione per AvgTimeOnQ (703) e QmaxDepth (739)

Utilizzare questo argomento per spiegare in che modo vengono trovate le variabili utilizzate nell'esempio per il monitoraggio dei dati delle statistiche.

Riepilogo:

- 703 indica MQIAMO64_AVG_Q_TIME che è **AvgTimeOnQ**
- 739 indica MQIAMO_Q_MAX_DEPTH che è **QMaxDepth**

Per individuare la mappatura è necessario eseguire due procedure:

1. Visitare “Dati messaggio statistiche coda” a pagina 194 per una spiegazione degli attributi “AvgTimeOnQ” a pagina 198 e “QMaxDepth” a pagina 197 man mano che si selezionano le *statistiche* per il parametro **Type** nel comando **amqsmn** .

In ogni caso, ricercare l'attributo **Identifier**.

Identifier per **AvgTimeOnQ** è MQIAMO64_AVG_Q_TIME e per **QMaxDepth** è MQIAMO_Q_MAX_DEPTH.

2. Ricercare gli ID trovati nel passo “1” a pagina 157.

Andare alla sezione Costanti e scorrere verso il basso finché non si trova l'elenco MQIAMO_* (Formato del comando Tipi di parametri di monitoraggio numero intero) . Individuare MQIAMO_Q_MAX_DEPTH e visualizzare il valore 739.

Allo stesso modo, individuare l'elenco MQIAMO64_* (Tipi di parametri di monitoraggio del formato del comando a 64 bit) . Individua MQIAMO64_AVG_Q_TIME e vedrai il valore 703.

Riferimento ai messaggi di statistiche e account

Utilizzare questa pagina per ottenere una panoramica del formato dei messaggi di account e statistiche e delle informazioni restituite in questi messaggi

I messaggi di statistiche e di account sono messaggi IBM MQ standard contenenti un descrittore di messaggi e dati di messaggi. I dati del messaggio contengono informazioni sulle operazioni MQI eseguite dalle applicazioni IBM MQ o sulle attività che si verificano in un sistema IBM MQ .

Descrittore messaggio

- Una struttura MQMD

Dati messaggio

- Un'intestazione PCF (MQCFH)
- I dati del messaggio di statistiche o di account che vengono sempre restituiti
- Dati del messaggio di statistiche o di account restituiti se disponibili

Formato del messaggio di statistiche e account

Utilizzare questa pagina come esempio della struttura di un messaggio di account MQI

Tabella 23. Struttura del messaggio di account MQI

Struttura MQMD	Struttura MQCFH intestazione messaggio account	Dati del messaggio di account MQI ¹
Identificativo struttura Versione della struttura Opzioni di documentazione Tipo messaggio Ora di scadenza Codice feedback Codifica ID set di caratteri codificato Formato del messaggio Priorità messaggio Persistenza ID messaggio Identificativo di correlazione Conteggio backout Coda di risposta Gestore code di risposta Identificativo utente Token account Dati identità applicazione Tipo di applicazione Nome applicazione Data Put Ora Put Dati origine applicazione ID gruppo Numero sequenza messaggi Offset Indicatori di messaggio Lunghezza originale	Tipo di struttura Lunghezza struttura Versione della struttura Identificativo comando Numero sequenza messaggi Opzioni di controllo Codice di completamento Codice di errore Conteggio parametro	Gestore code Data inizio intervallo Ora di inizio intervallo Data fine intervallo Ora di fine intervallo Livello di comandi Identificativo connessione Numero di sequenza Nome applicazione Identificativo processo applicazione Identificativo thread applicazione Identificativo utente Data di connessione Tempo di connessione Nome connessione Nome canale Data di disconnessione Tempo di disconnessione Tipo di disconnessione Conteggio aperture Numero errori di apertura Numero chiusure Numero errori di chiusura Numero immissioni Numero errori PUT Numero PUT1 Numero errori PUT1 Byte di immissione Ottieni conteggio Numero errori di ricezione Byte di ricezione Conteggio ricerche Numero errori di ricerca Byte di ricerca Conteggio commit Numero errori di commit Conteggio backout Numero interrogazioni Numero errori di interrogazione Numero impostazioni Numero errori di impostazione
<p>Nota:</p> <p>1. I parametri visualizzati sono quelli restituiti per un messaggio di account MQI. I dati effettivi del messaggio di statistiche o di account dipendono dalla categoria del messaggio.</p>		

Messaggio di account e statistiche MQMD (descrittore messaggio)

Utilizzare questa pagina per comprendere le differenze tra il descrittore di messaggi di account e statistiche e il descrittore di messaggi di eventi

I parametri e i valori nel descrittore del messaggio di account e statistiche sono gli stessi del descrittore del messaggio dei messaggi di eventi, con la seguente eccezione:

Format

Descrizione:	Nome formato dei dati del messaggio.
Tipo di dati:	MQCHAR8.
Valore:	MMQFMT_ADMIN Messaggio di amministrazione.

Alcuni dei parametri contenuti nel messaggio descrittore di account e statistiche contengono dati fissi forniti dal gestore code che ha generato il messaggio.

MQMD specifica anche il nome del gestore code (troncato a 28 caratteri) che inserisce il messaggio e la data e l'ora in cui il messaggio è stato inserito nella coda di account o statistiche.

Dati dei messaggi nei messaggi di statistiche e account

I dati del messaggio nei messaggi di account e statistiche si basano sul formato del comando programmabile (PCF), che viene utilizzato nelle richieste e risposte del comando PCF. I dati del messaggio nei messaggi di statistiche e account sono costituiti da un'intestazione PCF (MQCFH) e da un report di statistiche o account.

Messaggio statistiche e account MQCFH (intestazione PCF)

L'intestazione del messaggio dei messaggi di account e statistiche è una struttura MQCFH. I parametri e i valori nell'intestazione del messaggio del messaggio di account e statistiche sono gli stessi dell'intestazione del messaggio dei messaggi di evento, con le seguenti eccezioni:

Command

Descrizione:	Identificativo comando. Identifica la categoria di messaggi di account o statistiche.
Tipo di dati:	MQLONG.
Valori:	MQCMD_ACCOUNTING_MQI Messaggio di account MQI. MQCMD_XX_ENCODE_CASE_ONE conteggio_coda Messaggio di account coda. MQCMD_STATISTICS_MQI Messaggio statistiche MQI. MQCMD_STATISTICS_Q Messaggio statistiche coda. MQCMD_STATISTICS_CHALLEGATO Messaggio statistiche canale.

Version

Descrizione:	Numero di versione della struttura.
Tipo di dati:	MQLONG.
Valore:	MQCFH_VERSION_3 Version-3 per i messaggi di account e statistiche.

Dati dei messaggi di statistiche e account

Il contenuto dei dati del messaggio di statistiche e di account dipende dalla categoria del messaggio di statistiche o di account, come segue:

Messaggio di account MQI

I dati del messaggio di account MQI sono costituiti da un numero di parametri PCF, ma non da gruppi PCF.

Messaggio di account coda

I dati del messaggio di account della coda sono costituiti da un numero di parametri PCF e sono compresi nell'intervallo tra 1 e 100 *QAccountingData* gruppi PCF.

Messaggio statistiche MQI

I dati del messaggio delle statistiche MQI sono costituiti da un numero di parametri PCF, ma non da gruppi PCF.

Messaggio statistiche coda

I dati dei messaggi delle statistiche della coda sono costituiti da un numero di parametri PCF e nell'intervallo compreso tra 1 e 100 *QStatisticsData* gruppi PCF.

Messaggio statistiche canale

I dati dei messaggi delle statistiche dei canali sono costituiti da un numero di parametri PCF e sono compresi tra 1 e 100 *ChlStatisticsData* gruppi PCF.

Dati del messaggio di account MQI

Utilizzare questa pagina per visualizzare la struttura di un messaggio di account MQI

Nome messaggio:	Messaggio di account MQI.
Piattaforme:	Tutti, tranne IBM MQ for z/OS.
Coda di sistema:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

Descrizione:	Il nome del gestore code
Identificativo:	MQCA_Q_MGR_NAME
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_
Restituito:	Sempre

IntervalStartDate

Descrizione:	La data di inizio del periodo di controllo
Identificativo:	DATA STAR MQCAMO
Tipo di dati:	MQCFST
Lunghezza massima:	MQ_DATE_LENGTH
Restituito:	Sempre

IntervalStartTime

Descrizione:	L'ora di avvio del periodo di monitoraggio
Identificativo:	ORA_START_MQCAMO
Tipo di dati:	MQCFST

Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre

IntervalEndDate

Descrizione: La data di fine del periodo di monitoraggio
Identificativo: DATA_FINE MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Sempre

IntervalEndTime

Descrizione: L'ora di fine del periodo di monitoraggio
Identificativo: ORA_END_MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre

CommandLevel

Descrizione: Il livello di comando del gestore code
Identificativo: LIVELLO_COMMAND_MQI
Tipo di dati: MQCFIN
Restituito: Sempre

ConnectionId

Descrizione: L'identificativo di connessione per la connessione IBM MQ
Identificativo: ID_CONNESSIONE_MQBACF
Tipo di dati: MQCFB
Lunghezza massima: LUNGHEZZA_ID_CONNESSIONE_MQ
Restituito: Sempre

SeqNumber

Descrizione: Il numero di sequenza. Questo valore viene incrementato per ogni successivo record per connessioni di lunga durata.
Identificativo: NUMERO_SEQUENZA_MQIACF_
Tipo di dati: MQCFIN
Restituito: Sempre

ApplicationName

Descrizione:	Il nome dell'applicazione. Il contenuto di questo campo è equivalente al contenuto del campo <i>PutApplName</i> nella descrizione del messaggio.
Identificativo:	MQCACF_APPL_NAME
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_APPL_MQ
Restituito:	Sempre

ApplicationPid

Descrizione:	L'identificativo del processo del sistema operativo dell'applicazione
Identificativo:	ID_PROCESSO_MQIACF
Tipo di dati:	MQCFIN
Restituito:	Sempre

ApplicationTid

Descrizione:	L'identificativo del thread IBM MQ della connessione nell'applicazione
Identificativo:	ID_THREAD_MQIAC
Tipo di dati:	MQCFIN
Restituito:	Sempre

UserId

Descrizione:	Il contesto dell'identificativo utente dell'applicazione
Identificativo:	IDENTIFICATIVO_UTENTE_MQCACF
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_ID_UTENTE_MQ
Restituito:	Sempre

ConnDate

Descrizione:	Data dell'operazione MQCONN
Identificativo:	DATA_COLLEGA_MQCAMO
Tipo di dati:	MQCFST
Lunghezza massima:	MQ_ORA_LENGTH
Restituito:	Quando disponibile

ConnTime

Descrizione:	Ora dell'operazione MQCONN
Identificativo:	ORA_COLLEGA_MQCAMO
Tipo di dati:	MQCFST

Lunghezza massima: MQ_ORA_LENGTH
Restituito: Quando disponibile

ConnName

Descrizione: Nome connessione per connessione client
Identificativo: MQCACH_CONNECTION_NAME
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_CONN_MQ
Restituito: Quando disponibile

ChannelName

Descrizione: Nome canale per la connessione client
Identificativo: MQCACH_CHANNEL_NAME
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_CANALE_MQ
Restituito: Quando disponibile

RemoteProduct

Descrizione: Identificativo del prodotto remoto per la connessione client come visualizzato nel campo **RPRODUCT** di DISPLAY CHSATUS
Identificativo: MQCACH_REMOTE_PRODUCT
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_PRODOTTO_REMOTO_MQ_
Restituito: Quando disponibile

RemoteVersion

Descrizione: Versione del prodotto remoto per la connessione client come visualizzata nel campo **RVERSION** di DISPLAY CHSTATUS
Identificativo: MQCACH_REMOTE_VERSIONE
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_VERSIONE_REMOTA_MQ_
Restituito: Quando disponibile

DiscDate

Descrizione: Data dell'operazione MQDISC
Identificativo: DATA_DISCO MQCAMO
Tipo di dati: MQCFST

Lunghezza massima: MQ_DATE_LENGTH
Restituito: Quando disponibile

DiscTime

Descrizione: Ora dell'operazione MQDISC
Identificativo: TEMPO_DIS_CAMPO_DI_TABELLA
Tipo di dati: MQCFST
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Quando disponibile

DiscType

Descrizione: Tipo di disconnessione
Identificativo: TIPO_DISC_MQIAMO
Tipo di dati: MQCFIN
Valori: I valori possibili sono:
MQDISCONNECT_NORMALE
Richiesto dall'applicazione
MQDISCONNECT_IMPLICIT
Fine anomala dell'applicazione
MQDISCONNECT_Q_MGR
Connessione interrotta dal gestore code
Restituito: Quando disponibile

OpenCount

Descrizione: Il numero di oggetti aperti correttamente, emettendo direttamente una chiamata a MQOPEN o utilizzando il comando MQPUT1 . Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).
Identificativo: MQIAMO_APERTURE
Tipo di dati: MQCFIL
Restituito: Quando disponibile

OpenFailCount

Descrizione: Il numero di tentativi non riusciti di aprire un oggetto. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).
Identificativo: MQIAMO_OPENS_FAILED
Tipo di dati: MQCFIL
Restituito: Quando disponibile

CloseCount

Descrizione: Il numero di oggetti chiusi. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_CHIUSO
Tipo di dati: MQCFIL
Restituito: Quando disponibile

CloseFailCount

Descrizione: Il numero di tentativi non riusciti di chiudere un oggetto. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_CLOSES_FAILED
Tipo di dati: MQCFIL
Restituito: Quando disponibile

PutCount

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente in una coda, ad eccezione dei messaggi inseriti utilizzando la chiamata MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUTS
Tipo di dati: MQCFIL
Restituito: Quando disponibile

PutFailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio

Identificativo: MQIAMO_PUTS_FAILED
Tipo di dati: MQCFIN
Restituito: Quando disponibile

Put1Count

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente nella coda utilizzando chiamate MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUT1S
Tipo di dati: MQCFIL
Incluso nel gruppo *QAccountingData*
PCF:
Restituito: Quando disponibile

Put1FailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio utilizzando chiamate MQPUT1

Identificativo: MQIAMO_PUT1S_FAILED
Tipo di dati: MQCFIN

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

PutBytes

Descrizione: Il numero di byte scritti utilizzando chiamate di inserimento per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO64_PUT_BYTES

Tipo di dati: MQCFIL64

Restituito: Quando disponibile

GetCount

Descrizione: Il numero di chiamate MQGET distruttive riuscite per messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: GET_MQIAMO

Tipo di dati: MQCFIL

Restituito: Quando disponibile

GetFailCount

Descrizione: Il numero di chiamate MQGET distruttive non riuscite

Identificativo: MQIAMO_GETS_NON RIUSCITO

Tipo di dati: MQCFIN

Restituito: Quando disponibile

GetBytes

Descrizione: Numero totale di byte richiamati per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO64_GET_BYTES

Tipo di dati: MQCFIL64

Restituito: Quando disponibile

BrowseCount

Descrizione: Il numero di chiamate MQGET non distruttive riuscite per messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_BROWSES

Tipo di dati: MQCFIL

Restituito: Quando disponibile

BrowseFailCount

Descrizione: Il numero di chiamate MQGET non distruttive non riuscite

Identificativo: MQIAMO_BROWSES_FAILED
Tipo di dati: MQCFIN
Restituito: Quando disponibile

BrowseBytes

Descrizione: Numero totale di byte ricerche messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Identificativo: MQIAMO64_BROWSE_BYTES
Tipo di dati: MQCFIL64
Restituito: Quando disponibile

CommitCount

Descrizione: Il numero di transazioni riuscite. Questo valore include le transazioni sottoposte a commit implicitamente dall'applicazione connessa. Le richieste di commit in cui non è presente alcun lavoro in sospeso vengono incluse in questo conteggio.
Identificativo: COMMIT MQIAMO
Tipo di dati: MQCFIN
Restituito: Quando disponibile

CommitFailCount

Descrizione: Il numero di tentativi non riusciti di completare una transazione
Identificativo: MQIAMO_COMMITS_FAILED
Tipo di dati: MQCFIN
Restituito: Quando disponibile

BackCount

Descrizione: Il numero di backout elaborati, inclusi i backout impliciti a causa di una disconnessione anomala
Identificativo: BACKOUTS MQIAMO
Tipo di dati: MQCFIN
Restituito: Quando disponibile

InqCount

Descrizione: Il numero di oggetti con esito positivo sottoposti a interrogazione. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).
Identificativo: INQS MQIAMO
Tipo di dati: MQCFIL
Restituito: Quando disponibile

InqFailCount

Descrizione:	Il numero di tentativi di interrogazione oggetti non riusciti. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare Nota di riferimento 1 .
Identificativo:	MQIAMO_INQS_NON RIUSCITO
Tipo di dati:	MQCFIL
Restituito:	Quando disponibile

SetCount

Descrizione:	Il numero di chiamate MQSET riuscite. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare Nota di riferimento 1 .
Identificativo:	SETS MQIAMO
Tipo di dati:	MQCFIL
Restituito:	Quando disponibile

SetFailCount

Descrizione:	Il numero di chiamate MQSET non riuscite. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare Nota di riferimento 1 .
Identificativo:	MQIAMO_SETS_NON RIUSCITO
Tipo di dati:	MQCFIL
Restituito:	Quando disponibile

SubCountDur

Descrizione:	Il numero di richieste di sottoscrizione riuscite che hanno creato, modificato o ripreso le sottoscrizioni durevoli. Questo è un array di valori indicizzati in base al tipo di operazione 0 = Il numero di sottoscrizioni create 1 = Il numero di sottoscrizioni modificate 2 = Il numero di sottoscrizioni riprese
Identificativo:	MQIAMO_SUBS_DUR
Tipo di dati:	MQCFIL
Restituito:	Quando disponibile.

SubCountNDur

Descrizione:	Il numero di richieste di sottoscrizione riuscite che hanno creato, modificato o ripreso sottoscrizioni non durevoli. Questo è un array di valori indicizzati in base al tipo di operazione 0 = Il numero di sottoscrizioni create 1 = Il numero di sottoscrizioni modificate 2 = Il numero di sottoscrizioni riprese
Identificativo:	MQIAMO_SUBS_NDUR
Tipo di dati:	MQCFIL

Restituito: Quando disponibile.

SubFailCount

Descrizione: Il numero di richieste di sottoscrizione non riuscite.

Identificativo: MQIAMO_SUBS_NON RIUSCITO

Tipo di dati: MQCFIN

Restituito: Quando disponibile.

UnsubCountDur

Descrizione: Il numero di richieste di annullamento della sottoscrizione riuscite per sottoscrizioni durevoli. Questo è un array di valori indicizzati in base al tipo di operazione

0 - La sottoscrizione è stata chiusa ma non rimossa

1 - La sottoscrizione è stata chiusa e rimossa

Identificativo: MQIAMO_UNSUBS_DUR

Tipo di dati: MQCFIL

Restituito: Quando disponibile.

UnsubCountNDur

Descrizione: Il numero di richieste di annullamento della sottoscrizione riuscite per sottoscrizioni durevoli. Questo è un array di valori indicizzati in base al tipo di operazione

0 - La sottoscrizione è stata chiusa ma non rimossa

1 - La sottoscrizione è stata chiusa e rimossa

Identificativo: MQIAMO_UNSUBS_NDUR

Tipo di dati: MQCFIL

Restituito: Quando disponibile.

UnsubFailCount

Descrizione: Il numero di richieste di annullamento della sottoscrizione non riuscite.

Identificativo: MQIAMO_UNSUBS_NON RIUSCITO

Tipo di dati: MQCFIN

Restituito: Quando disponibile.

SubRqCount

Descrizione: Il numero di richieste MQSUBRQ riuscite.

Identificativo: MQIAMO_SUBRQS

Tipo di dati: MQCFIN

Restituito: Quando disponibile.

SubRqFailCount

Descrizione: Il numero di richieste MQSUB non riuscite.

Identificativo: MQIAMO_SUBRQS_NON RIUSCITO
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

CBCount

Descrizione: Il numero di richieste MQCB riuscite. Questo è un array di valori indicizzati in base al tipo di operazione
0 - Un callback è stato creato o modificato
1 - È stato rimosso un callback
2 - È stata ripresa una richiamata
3 - Un callback è stato sospeso

Identificativo: Cbs MQIAMO
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

CBFailCount

Descrizione: Il numero di richieste MQCB non riuscite.
Identificativo: MQIAMO_CBS_NON RIUSCITO
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

CtlCount

Descrizione: Il numero di richieste MQCTL riuscite. Questo è un array di valori indicizzati in base al tipo di operazione
0 - La connessione è stata avviata
1 - La connessione è stata arrestata
2 - La connessione è stata ripresa
3 - La connessione è stata sospesa

Identificativo: CTLS MQIAMO
Tipo di dati: MQCFIL
Restituito: Quando disponibile.

CtlFailCount

Descrizione: Il numero di richieste MQCTL non riuscite.
Identificativo: MQIAMO_CTLN_NON RIUSCITO
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

StatCount

Descrizione: Il numero di richieste MQSTAT riuscite.
Identificativo: MQIAMO_STATS.

Tipo di dati: MQCFIN
Restituito: Quando disponibile.

StatFailCount

Descrizione: Il numero di richieste MQSTAT non riuscite.
Identificativo: MQIAMO_STATS_NON RIUSCITO
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

PutTopicCount

Descrizione: Il numero di messaggi persistenti e non persistenti correttamente inseriti in un argomento, ad eccezione dei messaggi immessi utilizzando la chiamata MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Nota: i messaggi inseriti utilizzando un alias coda che si risolvono in un argomento sono inclusi in questo valore.
Identificativo: MQIAMO_TOPIC_PUTS
Tipo di dati: MQCFIL
Restituito: Quando disponibile.

PutTopicFailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio in un argomento.
Identificativo: MQIAMO_TOPIC_PUTS_FAILED
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

Put1TopicCount

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente in un argomento utilizzando chiamate MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Nota: i messaggi inseriti utilizzando un alias coda che si risolvono in un argomento sono inclusi in questo valore.
Identificativo: MQIAMO_TOPIC_PUT1S
Tipo di dati: MQCFIL
Restituito: Quando disponibile.

Put1TopicFailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio in un argomento utilizzando chiamate MQPUT1 .
Identificativo: MQIAMO_TOPIC_PUT1S_FAILED
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

PutTopicBytes

Descrizione:	Il numero di byte scritti utilizzando le chiamate di inserimento per i messaggi persistenti e non persistenti che si risolvono in un'operazione di pubblicazione. Questo è il numero di byte immessi dall'applicazione e non il numero risultante di byte consegnati ai sottoscrittori. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_TOPIC_PUT_BYTES
Tipo di dati:	MQCFIL64
Restituito:	Quando disponibile.

Dati del messaggio di account coda

Utilizzare questa pagina per visualizzare la struttura di un messaggio di account coda

Nome messaggio:	Messaggio di account coda.
Piattaforme:	Tutti, tranne IBM MQ for z/OS.
Coda di sistema:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

Descrizione:	Il nome del gestore code
Identificativo:	MQCA_Q_MGR_NAME
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_
Restituito:	Sempre

IntervalStartDate

Descrizione:	La data di inizio del periodo di controllo
Identificativo:	DATA STAR MQCAMO
Tipo di dati:	MQCFST
Lunghezza massima:	MQ_DATE_LENGTH
Restituito:	Sempre

IntervalStartTime

Descrizione:	L'ora di avvio del periodo di monitoraggio
Identificativo:	ORA_START_MQCAMO
Tipo di dati:	MQCFST
Lunghezza massima:	MQ_ORA_LENGTH
Restituito:	Sempre

IntervalEndDate

Descrizione:	La data di fine del periodo di monitoraggio
--------------	---

Identificativo: DATA_FINE MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Sempre

IntervalEndTime

Descrizione: L'ora di fine del periodo di monitoraggio
Identificativo: ORA_END_MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre

CommandLevel

Descrizione: Il livello di comando del gestore code
Identificativo: LIVELLO_COMMAND_MQI
Tipo di dati: MQCFIN
Restituito: Sempre

ConnectionId

Descrizione: L'identificativo di connessione per la connessione IBM MQ
Identificativo: ID_CONNESSIONE_MQBACF
Tipo di dati: MQCFB
Lunghezza massima: LUNGHEZZA_ID_CONNESSIONE_MQ
Restituito: Sempre

SeqNumber

Descrizione: Il numero di sequenza. Questo valore viene incrementato per ogni successivo record per connessioni di lunga durata.
Identificativo: NUMERO_SEQUENZA_MQIACF_
Tipo di dati: MQCFIN
Restituito: Sempre

ApplicationName

Descrizione: Il nome dell'applicazione. Il contenuto di questo campo è equivalente al contenuto del campo Nome PutAppl nel descrizione del messaggio.
Identificativo: MQCACF_APPL_NAME
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_APPL_MQ
Restituito: Sempre

ApplicationPid

Descrizione:	L'identificativo del processo del sistema operativo dell'applicazione
Identificativo:	ID_PROCESSO MQIACF
Tipo di dati:	MQCFIN
Restituito:	Sempre

ApplicationTid

Descrizione:	L'identificativo del thread IBM MQ della connessione nell'applicazione
Identificativo:	ID_THREAD_MQIAC
Tipo di dati:	MQCFIN
Restituito:	Sempre

UserId

Descrizione:	Il contesto dell'identificativo utente dell'applicazione
Identificativo:	IDENTIFICATIVO_UTENTE_MQCACF
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_ID_UTENTE_MQ
Restituito:	Sempre

ChannelName

Descrizione:	Nome canale per la connessione client
Identificativo:	MQCACH_CHANNEL_NAME
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_CANALE_MQ
Restituito:	Quando disponibile

ConnName

Descrizione:	Nome connessione per connessione client
Identificativo:	MQCACH_CONNECTION_NAME
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_CONN_MQ
Restituito:	Quando disponibile

ObjectCount

Descrizione:	Il numero di code a cui è stato eseguito l'accesso nell'intervallo per cui sono stati registrati i dati di account. Questo valore è impostato sul numero di gruppi PCF <i>QAccountingData</i> contenuti nel messaggio.
Identificativo:	CONTEGGIO_OGGETTI_MQIAMO_
Tipo di dati:	MQCFIN

Restituito: Sempre

QAccountingData

Descrizione: Parametri raggruppati che specificano i dettagli di account per una coda

Identificativo: DATI_CONTEGGIO_Q_MQGACF

Tipo di dati: MQCFGR

Parametri nel gruppo:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- OpenCount*
- OpenDate*
- OpenTime*
- CloseDate*
- CloseTime*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- PutMinBytes*
- PutMaxBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- GetMinBytes*
- GetMaxBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- BrowseMinBytes*
- BrowseMaxBytes*
- TimeOnQMin*
- TimeOnQAvg*
- TimeOnQMax*

Restituito: Sempre

QName

Descrizione: Il nome della coda

Identificativo: Nome_Q_MQCA

Tipo di dati: MQCFST

Incluso nel gruppo *QAccountingData*
PCF:

Lunghezza massima: LUNGHEZZA_NOME_Q_MQ

Restituito: Quando disponibile

CreateDate

Descrizione: La data di creazione della coda
Identificativo: DATA CREAZIONE MQCA
Tipo di dati: MQCFST
Incluso nel gruppo *QAccountingData*
PCF:
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Quando disponibile

CreateTime

Descrizione: L'ora di creazione della coda
Identificativo: ORA_MQCA_CREAZIONE
Tipo di dati: MQCFST
Incluso nel gruppo *QAccountingData*
PCF:
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Quando disponibile

QType

Descrizione: Il tipo della coda
Identificativo: TIPO_Q_MQI
Tipo di dati: MQCFIN
Incluso nel gruppo *QAccountingData*
PCF:
Valore: LOCALE MQQT
Restituito: Quando disponibile

QDefinitionType

Descrizione: Il tipo di definizione della coda
Identificativo: TIPO_DI_DEFINIZIONE MQIA
Tipo di dati: MQCFIN
Incluso nel gruppo *QAccountingData*
PCF:
Valori: I possibili valori sono:
MQQDT_PREDEFINED
MQQDT_PERMANENT_DYNAMIC
MQQDT_TEMPORARY_DYNAMIC
Restituito: Quando disponibile

OpenCount

Descrizione: Il numero di volte in cui questa coda è stata aperta dall'applicazione in questo intervallo, emettendo direttamente una chiamata a MQOPEN o utilizzando il verbo MQPUT1 .

Identificativo: MQIAMO_APERTURE

Tipo di dati: MQCFIL

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

OpenDate

Descrizione: La data in cui la coda è stata aperta per la prima volta in questo intervallo di registrazione. Se la coda era già aperta all'avvio di questo intervallo, questo valore riflette la data in cui la coda è stata originariamente aperta.

Identificativo: DATA_OPEN_MQCAMO

Tipo di dati: MQCFST

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

OpenTime

Descrizione: L'ora in cui la coda è stata aperta per la prima volta in questo intervallo di registrazione. Se la coda era già aperta all'inizio di questo intervallo, questo valore riflette l'ora in cui la coda è stata originariamente aperta.

Identificativo: ORA_OPEN_MQCAMO

Tipo di dati: MQCFST

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

CloseDate

Descrizione: La data della chiusura finale della coda in questo intervallo di registrazione. Se la coda è ancora aperta, il valore non viene restituito.

Identificativo: DATA_CLOSE_MQCAMO

Tipo di dati: MQCFST

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

CloseTime

Descrizione: L'ora di chiusura finale della coda in questo intervallo di registrazione. Se la coda è ancora aperta, il valore non viene restituito.

Identificativo: ORA_CLOSE_MQCAMO

Tipo di dati: MQCFST

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

PutCount

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente nella coda, ad eccezione delle chiamate MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUTS

Tipo di dati: MQCFIL

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

PutFailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio, ad eccezione delle chiamate MQPUT1

Identificativo: MQIAMO_PUTS_FAILED

Tipo di dati: MQCFIN

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

Put1Count

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente nella coda utilizzando chiamate MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUT1S

Tipo di dati: MQCFIL

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

Put1FailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio utilizzando chiamate MQPUT1

Identificativo: MQIAMO_PUT1S_FAILED

Tipo di dati: MQCFIN

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

PutBytes

Descrizione:	Il numero totale di byte immessi per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_PUT_BYTES
Tipo di dati:	MQCFIL64
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

PutMinBytes

Descrizione:	La dimensione minima dei messaggi persistenti e non persistenti collocati nella coda. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO_PUT_MIN_BYTES
Tipo di dati:	MQCFIL
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

PutMaxBytes

Descrizione:	La dimensione massima dei messaggi persistenti e non persistenti collocati nella coda. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO_PUT_MAX_BYTES
Tipo di dati:	MQCFIL
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

GeneratedMsgCount

Descrizione:	Il numero di messaggi generati. I messaggi generati sono <ul style="list-style-type: none">• Eventi di profondità coda elevata• Eventi di grandezza coda bassa
Identificativo:	MQIAMO_GENERATED_MSGS
Tipo di dati:	MQCFIN
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

GetCount

Descrizione:	Il numero di chiamate MQGET distruttive riuscite per messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
--------------	---

Identificativo: GET_MQIAMO
Tipo di dati: MQCFIL
Incluso nel gruppo *QAccountingData*
PCF:
Restituito: Quando disponibile

GetFailCount

Descrizione: Il numero di chiamate MQGET distruttive non riuscite
Identificativo: MQIAMO_GETS_NON_RIUSCITO
Tipo di dati: MQCFIN
Incluso nel gruppo *QAccountingData*
PCF:
Restituito: Quando disponibile

GetBytes

Descrizione: Il numero di byte letti nelle chiamate MQGET distruttive per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Identificativo: MQIAMO64_GET_BYTES
Tipo di dati: MQCFIL64
Incluso nel gruppo *QAccountingData*
PCF:
Restituito: Quando disponibile

GetMinBytes

Descrizione: La dimensione del messaggio persistente e non persistente più piccolo richiamato dalla coda. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Identificativo: MQIAMO_GET_MIN_BYTES
Tipo di dati: MQCFIL
Incluso nel gruppo *QAccountingData*
PCF:
Restituito: Quando disponibile

GetMaxBytes

Descrizione: La dimensione del messaggio persistente e non persistente più grande richiamato dalla coda. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Identificativo: MQIAMO_GET_MAX_BYTES
Tipo di dati: MQCFIL
Incluso nel gruppo *QAccountingData*
PCF:
Restituito: Quando disponibile

BrowseCount

Descrizione:	Il numero di chiamate MQGET non distruttive riuscite per messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO_BROWSES
Tipo di dati:	MQCFIL
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

BrowseFailCount

Descrizione:	Il numero di chiamate MQGET non distruttive non riuscite
Identificativo:	MQIAMO_BROWSES_FAILED
Tipo di dati:	MQCFIN
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

BrowseBytes

Descrizione:	Il numero di byte letti in chiamate MQGET non distruttive che hanno restituito messaggi persistenti
Identificativo:	MQIAMO64_BROWSE_BYTES
Tipo di dati:	MQCFIL64
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

BrowseMinBytes

Descrizione:	La dimensione del messaggio persistente e non persistente più piccolo analizzato dalla coda. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO_BROWSE_MIN_BYTES
Tipo di dati:	MQCFIL
Incluso nel gruppo PCF:	<i>QAccountingData</i>
Restituito:	Quando disponibile

BrowseMaxBytes

Descrizione:	La dimensione del messaggio persistente e non persistente più grande esplorato dalla coda. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO_BROWSE_MAX_BYTES
Tipo di dati:	MQCFIL

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

TimeOnQMin

Descrizione: Il tempo più breve per cui un messaggio persistente e non persistente è rimasto nella coda prima di essere richiamato in modo distruttivo, in microsecondi. Per i messaggi richiamati nel punto di sincronizzazione, questo valore non include il tempo prima del commit dell'operazione get. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO64_Q_TIME_MIN

Tipo di dati: MQCFIL64

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

TimeOnQAvg

Descrizione: Il tempo medio in cui un messaggio persistente e non persistente è rimasto nella coda prima di essere richiamato in modo distruttivo, in microsecondi. Per i messaggi richiamati nel punto di sincronizzazione, questo valore non include il tempo prima del commit dell'operazione get. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO64_Q_TIME_AVG

Tipo di dati: MQCFIL64

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

TimeOnQMax

Descrizione: Il tempo più lungo per cui un messaggio persistente e non persistente è rimasto nella coda prima di essere richiamato in modo distruttivo, in microsecondi. Per i messaggi richiamati nel punto di sincronizzazione, questo valore non include il tempo prima del commit dell'operazione get. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO64_Q_TIME_MAX

Tipo di dati: MQCFIL64

Incluso nel gruppo *QAccountingData*
PCF:

Restituito: Quando disponibile

CBCount

Descrizione:	Il numero di richieste MQCB riuscite. Questo è un array di valori indicizzati in base al tipo di operazione 0 - Un callback è stato creato o modificato 1 - È stato rimosso un callback 2 - È stata ripresa una richiamata 3 - Un callback è stato sospeso
Identificativo:	Cbs MQIAMO
Tipo di dati:	MQCFIN
Restituito:	Quando disponibile.

CBFailCount

Descrizione:	Il numero di richieste MQCB non riuscite.
Identificativo:	MQIAMO_CBS_NON RIUSCITO
Tipo di dati:	MQCFIN
Restituito:	Quando disponibile.

Dati del messaggio delle statistiche MQI

Utilizzare questa pagina per visualizzare la struttura di un messaggio di statistiche MQI

Nome messaggio:	Messaggio statistiche MQI.
Piattaforme:	Tutti, tranne IBM MQ for z/OS.
Coda di sistema:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Descrizione:	Il nome del gestore code.
Identificativo:	MQCA_Q_MGR_NAME.
Tipo di dati:	MQCFST.
Lunghezza massima:	MQ_Q_MGR_NAME_LENGTH.
Restituito:	Sempre.

IntervalStartDate

Descrizione:	La data all'inizio del periodo di controllo.
Identificativo:	MQCAMO_START_DATE.
Tipo di dati:	MQCFST.
Lunghezza massima:	MQ_DATE_LENGTH
Restituito:	Sempre.

IntervalStartTime

Descrizione:	L'ora all'inizio del periodo di controllo.
Identificativo:	MQCAMO_START_TIME.

Tipo di dati: MQCFST.
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre.

IntervalEndDate

Descrizione: La data alla fine del periodo di controllo.
Identificativo: DATA_END_CAMO.
Tipo di dati: MQCFST.
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Sempre.

IntervalEndTime

Descrizione: L'ora alla fine del periodo di controllo.
Identificativo: MQCAMO_END_TIME.
Tipo di dati: MQCFST.
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre.

CommandLevel

Descrizione: Il livello di comando del gestore code.
Identificativo: MQIA_COMMAND_LEVEL.
Tipo di dati: MQCFIN.
Restituito: Sempre.

ConnCount

Descrizione: Il numero di connessioni al gestore code riuscite.
Identificativo: MQIAMO_CONNS.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

ConnFailCount

Descrizione: Il numero di tentativi di connessione non riusciti.
Identificativo: MQIAMO_CONNS_FAILED.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

ConnsMax

Descrizione: Il numero massimo di connessioni simultanee nell'intervallo di registrazione.
Identificativo: MQIAMO_CONNS_MAX.

Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

DiscCount

Descrizione: Il numero di disconnessioni dal gestore code. Si tratta di un array di numeri interi, indicizzati dalle seguenti costanti:

- MQDISCONNECT_NORMALE
- MQDISCONNECT_IMPLICIT
- MQDISCONNECT_Q_MGR

Identificativo: DISCO MQIAMO.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

OpenCount

Descrizione: Il numero di oggetti aperti correttamente, emettendo direttamente una chiamata a MQOPEN o utilizzando il comando MQPUT1 . Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_OPEN.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

OpenFailCount

Descrizione: Il numero di tentativi di apertura oggetti non riusciti. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_OPENS_FAILED.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

CloseCount

Descrizione: Il numero di oggetti chiusi correttamente. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_CHIUSURA.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

CloseFailCount

Descrizione: Il numero di tentativi di chiusura oggetti non riusciti. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_CLOSES_FAILED.
Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

InqCount

Descrizione: Il numero di oggetti interrogati correttamente. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_INQS.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

InqFailCount

Descrizione: Il numero di tentativi di interrogazione oggetti non riusciti. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_INQS_NON RIUSCITO.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

SetCount

Descrizione: Il numero di oggetti aggiornati correttamente (SET). Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_SETS.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

SetFailCount

Descrizione: Il numero di tentativi SET non riusciti. Questo parametro è un elenco di numeri interi indicizzato in base al tipo di oggetto, consultare [Nota di riferimento 1](#).

Identificativo: MQIAMO_SETS_NON RIUSCITO.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

PutCount

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente in una coda, ad eccezione delle richieste MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUTS.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

PutFailCount

Descrizione: Il numero di tentativi di messaggi di inserimento non riusciti.

Identificativo: MQIAMO_PUTS_FAILED.

Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

Put1Count

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente in una coda utilizzando richieste MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#)

Identificativo: MQIAMO_PUT1S.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

Put1FailCount

Descrizione: Il numero di tentativi non riusciti di immissione di un messaggio persistente e non persistente in una coda mediante richieste MQPUT1. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#)

Identificativo: MQIAMO_PUT1S_FAILED.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

PutBytes

Descrizione: Il numero di byte per i messaggi persistenti e non persistenti scritti utilizzando le richieste di inserimento. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#)

Identificativo: MQIAMO64_PUT_BYTES.
Tipo di dati: MQCFIL64.
Restituito: Quando disponibile.

GetCount

Descrizione: Il numero di richieste di ricezione distruttive riuscite per messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#)

Identificativo: MQIAMO_GETS.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

GetFailCount

Descrizione: Il numero di richieste di ricezione distruttive.

Identificativo: MQIAMO_GETS_NON RIUSCITO.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

GetBytes

Descrizione:	Il numero di byte letti nelle richieste di ricezione distruttive per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2
Identificativo:	MQIAMO64_GET_BYTES.
Tipo di dati:	MQCFIL64.
Restituito:	Quando disponibile.

BrowseCount

Descrizione:	Il numero di richieste di ricezione non distruttive riuscite per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2
Identificativo:	MQIAMO_BROWSES.
Tipo di dati:	MQCFIL.
Restituito:	Quando disponibile.

BrowseFailCount

Descrizione:	Il numero di richieste di ricezione non distruttive.
Identificativo:	MQIAMO_BROWSES_FAILED.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

BrowseBytes

Descrizione:	Il numero di byte letti nelle richieste di ricezione non distruttive per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2
Identificativo:	MQIAMO64_BROWSE_BYTES.
Tipo di dati:	MQCFIL64.
Restituito:	Quando disponibile.

CommitCount

Descrizione:	Il numero di transazioni completate correttamente. Questo numero include le transazioni di cui è stato eseguito il commit implicitamente dalla disconnessione dell'applicazione e le richieste di commit in cui non è presente alcun lavoro in sospenso.
Identificativo:	MQIAMO_COMMITS.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

CommitFailCount

Descrizione:	Il numero di tentativi non riusciti di completamento di una transazione.
Identificativo:	MQIAMO_COMMITS_FAILED.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

BackCount

Descrizione:	Il numero di backout elaborati, inclusi quelli impliciti in caso di disconnessione anomala.
Identificativo:	MQIAMO_BACKOUTS.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

ExpiredMsgCount

Descrizione:	Il numero di messaggi persistenti e non persistenti eliminati in quanto scaduti prima che potessero essere richiamati.
Identificativo:	MQIAMO_MSGS_EXPIRED.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

PurgeCount

Descrizione:	Il numero di volte in cui la coda è stata cancellata.
Identificativo:	MQIAMO_MSGS_PURGED.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

SubCountDur

Descrizione:	Il numero di richieste di sottoscrizione riuscite che hanno creato, modificato o ripreso le sottoscrizioni durevoli. Questo è un array di valori indicizzati in base al tipo di operazione 0 = Il numero di sottoscrizioni create 1 = Il numero di sottoscrizioni modificate 2 = Il numero di sottoscrizioni riprese
Identificativo:	MQIAMO_SUBS_DUR.
Tipo di dati:	MQCFIL
Restituito:	Quando disponibile.

SubCountNDur

Descrizione:	Il numero di richieste di sottoscrizione riuscite che hanno creato, modificato o ripristinato sottoscrizioni non durevoli. Questo è un array di valori indicizzati in base al tipo di operazione 0 = Il numero di sottoscrizioni create 1 = Il numero di sottoscrizioni modificate 2 = Il numero di sottoscrizioni riprese
Identificativo:	MQIAMO_SUBS_NDUR.
Tipo di dati:	MQCFIL.
Restituito:	Quando disponibile.

SubFailCount

Descrizione: Il numero di richieste di sottoscrizione non riuscite.
Identificativo: MQIAMO_SUBS_FAILED.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

UnsubCountDur

Descrizione: Il numero di richieste di annullamento della sottoscrizione riuscite per sottoscrizioni durevoli. Questo è un array di valori indicizzati in base al tipo di operazione
0 - La sottoscrizione è stata chiusa ma non rimossa
1 - La sottoscrizione è stata chiusa e rimossa
Identificativo: MQIAMO_UNSUBS_DUR.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

UnsubCountNDur

Descrizione: Il numero di richieste di annullamento della sottoscrizione riuscite per sottoscrizioni non durevoli. Questo è un array di valori indicizzati in base al tipo di operazione
0 - La sottoscrizione è stata chiusa ma non rimossa
1 - La sottoscrizione è stata chiusa e rimossa
Identificativo: MQIAMO_UNSUBS_NDUR.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

UnsubFailCount

Descrizione: Il numero di richieste di annullamento della sottoscrizione non riuscite.
Identificativo: MQIAMO_UNSUBS_FAILED.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

SubRqCount

Descrizione: Il numero di richieste MQSUBRQ riuscite.
Identificativo: MQIAMO_SUBRQS
Tipo di dati: MQCFIN
Restituito: Quando disponibile.

SubRqFailCount

Descrizione: Il numero di richieste MQSUBRQ non riuscite.
Identificativo: MQIAMO_SUBRQS_NON RIUSCITO.
Tipo di dati: MQCFIN.

Restituito: Quando disponibile.

CBCount

Descrizione: Il numero di richieste MQCB riuscite. Questo è un array di valori indicizzati in base al tipo di operazione

0 - Un callback è stato creato o modificato

1 - È stato rimosso un callback

2 - È stata ripresa una richiamata

3 - Un callback è stato sospeso

Identificativo: MQIAMO_CBS.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

CBFailCount

Descrizione: Il numero di richieste MQCB non riuscite.

Identificativo: MQIAMO_CBS_NON RIUSCITO.

Tipo di dati: MQCFIN.

Restituito: Quando disponibile.

CtlCount

Descrizione: Il numero di richieste MQCTL riuscite. Questo è un array di valori indicizzati in base al tipo di operazione:

0 - La connessione è stata avviata

1 - La connessione è stata arrestata

2 - La connessione è stata ripresa

3 - La connessione è stata sospesa

Identificativo: MQIAMO_CTL.

Tipo di dati: MQCFIL.

Restituito: Quando disponibile.

CtlFailCount

Descrizione: Il numero di richieste MQCTL non riuscite.

Identificativo: MQIAMO_CTL.

Tipo di dati: MQCFIN.

Restituito: Quando disponibile.

StatCount

Descrizione: Il numero di richieste MQSTAT riuscite.

Identificativo: MQIAMO_STATS.

Tipo di dati: MQCFIN.

Restituito: Quando disponibile.

StatFailCount

Descrizione: Il numero di richieste MQSTAT non riuscite.
Identificativo: MQIAMO_STATS_NON RIUSCITO.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

SubCountDurHighWater

Descrizione: Il limite massimo del numero di sottoscrizioni durevoli durante l'intervallo di tempo. Questa è una schiera di valori indicizzati da SUBTYPE
0 - Il limite massimo per tutte le sottoscrizioni durevoli nel sistema
1 - Il limite massimo per le sottoscrizioni dell'applicazione durevole (MQSUBTYPE_API)
2 - Il limite massimo per la sottoscrizione di gestione durevole (MQSUBTYPE_ADMIN)
3 - Il limite massimo per le sottoscrizioni proxy durevoli (MQSUBTYPE_PROXY)

Identificativo: MQIAMO_SUB_DUR_HIGHWATER
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

SubCountDurLowWater

Descrizione: Il limite minimo sul numero di sottoscrizioni durevoli durante l'intervallo di tempo. Questa è una schiera di valori indicizzati da SUBTYPE.
0 - Il limite minimo per tutte le sottoscrizioni durevoli nel sistema
1 - Il limite minimo per le sottoscrizioni dell'applicazione durevole (MQSUBTYPE_API)
2 - Il limite minimo per le sottoscrizioni di amministrazione durevoli (MQSUBTYPE_ADMIN)
3 - Il limite minimo per le sottoscrizioni proxy durevoli (MQSUBTYPE_PROXY)

Identificativo: MQIAMO_SUB_DUR_LOWWATER
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

SubCountNDurHighWater

Descrizione: Il limite massimo del numero di sottoscrizioni non durevoli durante l'intervallo di tempo. Questa è una schiera di valori indicizzati da SUBTYPE
0 - Il limite massimo per tutte le sottoscrizioni non durevoli nel sistema
1 - Il limite massimo per le sottoscrizioni di applicazioni non durevoli (MQSUBTYPE_API)
2 - Il limite massimo per la sottoscrizione di gestione non durevole (MQSUBTYPE_ADMIN)
3 - Il limite massimo per le sottoscrizioni proxy non durevoli (MQSUBTYPE_PROXY)

Identificativo: MQIAMO_SUB_NDUR_HIGHWATER

Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

SubCountNDurLowWater

Descrizione: Il limite minimo del numero di sottoscrizioni non durevoli durante l'intervallo di tempo. Questa è una schiera di valori indicizzati da SUBTYPE.
0 - Il limite minimo per tutte le sottoscrizioni non durevoli nel sistema
1 - Il limite minimo per le sottoscrizioni di applicazioni non durevoli (MQSUBTYPE_API)
2 - Il limite minimo per le sottoscrizioni admin non durevoli (MQSUBTYPE_ADMIN)
3 - Il limite minimo per le sottoscrizioni proxy non durevoli (MQSUBTYPE_PROXY)

Identificativo: MQIAMO_SUB_NDUR_LOWWATER
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

PutTopicCount

Descrizione: Il numero di messaggi persistenti e non persistenti correttamente inseriti in un argomento, ad eccezione dei messaggi immessi utilizzando la chiamata MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Nota: i messaggi inseriti utilizzando un alias coda che si risolvono in un argomento sono inclusi in questo valore.

Identificativo: MQIAMO_TOPIC_PUTS.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

PutTopicFailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio in un argomento.

Identificativo: MQIAMO_TOPIC_PUTS_FAILED.
Tipo di dati: MQCFIN.
Restituito: Quando disponibile.

Put1TopicCount

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente in un argomento utilizzando chiamate MQPUT1 . Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).
Nota: i messaggi inseriti utilizzando un alias coda che si risolvono in un argomento sono inclusi in questo valore.

Identificativo: MQIAMO_TOPIC_PUT1S.
Tipo di dati: MQCFIL.
Restituito: Quando disponibile.

Put1TopicFailCount

Descrizione:	Il numero di tentativi non riusciti di inserire un messaggio in un argomento utilizzando chiamate MQPUT1 .
Identificativo:	MQIAMO_TOPIC_PUT1S_FAILED.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

PutTopicBytes

Descrizione:	Il numero di byte scritti utilizzando le chiamate di inserimento per i messaggi persistenti e non persistenti che si risolvono in un'operazione di pubblicazione. Questo è il numero di byte immessi dall'applicazione e non il numero risultante di byte consegnati ai sottoscrittori, consultare PublishMsgByte per questo valore. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_TOPIC_PUT_BYTES.
Tipo di dati:	MQCFIL64.
Restituito:	Quando disponibile.

PublishMsgCount

Descrizione:	Il numero di messaggi consegnati alle sottoscrizioni nell'intervallo di tempo. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_PUBLISH_MSG_COUNT
Tipo di dati:	MQCFIL.
Restituito:	Quando disponibile.

PublishMsgBytes

Descrizione:	Il numero di byte recapitati alle sottoscrizioni nell'intervallo di tempo. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_PUBLISH_MSG_BYTES
Tipo di dati:	MQCFIL64.
Restituito:	Quando disponibile.

Dati messaggio statistiche coda

Utilizzare questa pagina per visualizzare la struttura di un messaggio di statistiche della coda

Nome messaggio:	Messaggio statistiche coda.
Piattaforme:	Tutti, tranne IBM MQ for z/OS.
Coda di sistema:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Descrizione:	Il nome del gestore code
Identificativo:	MQCA_Q_MGR_NAME
Tipo di dati:	MQCFST

Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_
Restituito: Sempre

IntervalStartDate

Descrizione: La data di inizio del periodo di monitoraggio
Identificativo: DATA_STAR_MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Sempre

IntervalStartTime

Descrizione: L'ora all'inizio del periodo di monitoraggio
Identificativo: ORA_START_MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre

IntervalEndDate

Descrizione: La data alla fine del periodo di monitoraggio
Identificativo: DATA_FINE_MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Sempre

IntervalEndTime

Descrizione: L'ora alla fine del periodo di monitoraggio
Identificativo: ORA_END_MQCAMO
Tipo di dati: MQCFST
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre

CommandLevel

Descrizione: Il livello di comando del gestore code
Identificativo: LIVELLO_COMMAND_MQI
Tipo di dati: MQCFIN
Restituito: Sempre

ObjectCount

Descrizione:	Il numero di oggetti coda a cui è stato eseguito l'accesso nell'intervallo per cui sono stati registrati i dati delle statistiche. Questo valore è impostato sul numero di gruppi PCF QStatisticsData contenuti nel messaggio.
Identificativo:	CONTEGGIO_OGGETTI_MQIAMO_
Tipo di dati:	MQCFIN
Restituito:	Sempre

QStatisticsData

Descrizione:	Parametri raggruppati che specificano i dettagli delle statistiche per una coda
Identificativo:	MQGACF_Q_STATISTICS_DATA
Tipo di dati:	MQCFGR
Parametri nel gruppo:	<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>
Restituito:	Sempre

QName

Descrizione:	Il nome della coda
Identificativo:	Nome_Q_MQCA
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ
Restituito:	Sempre

CreateDate

Descrizione:	La data di creazione della coda
--------------	---------------------------------

Identificativo: DATA CREAZIONE MQCA
Tipo di dati: MQCFST
Lunghezza massima: MQ_DATE_LENGTH
Restituito: Sempre

CreateTime

Descrizione: L'ora in cui la coda è stata creata
Identificativo: ORA_MQCA_CREAZIONE
Tipo di dati: MQCFST
Lunghezza massima: MQ_ORA_LENGTH
Restituito: Sempre

QType

Descrizione: Il tipo della coda
Identificativo: TIPO_Q_MQI
Tipo di dati: MQCFIN
Valore: LOCALE MQOTT
Restituito: Sempre

QDefinitionType

Descrizione: Il tipo di definizione della coda
Identificativo: TIPO_DI_DEFINIZIONE MQIA
Tipo di dati: MQCFIN
Valori: I valori possibili sono

- MQQDT_PREDEFINED
- MQQDT_PERMANENT_DYNAMIC
- MQQDT_TEMPORARY_DYNAMIC

Restituito: Quando disponibile

QMinDepth

Descrizione: La profondità minima della coda durante il periodo di controllo
Identificativo: MQIAMO_Q_MIN_DEPTH
Tipo di dati: MQCFIN
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

QMaxDepth

Descrizione: La profondità massima della coda durante il periodo di controllo
Identificativo: MQIAMO_Q_MAX_DEPTH

Tipo di dati: MQCFIN
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

AvgTimeOnQ

Descrizione: La latenza media, in microsecondi, dei messaggi richiamati in modo distruttivo dalla coda durante il periodo di controllo. Questo parametro è un elenco di numeri interi indicizzati in base al valore di persistenza, consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO64_AVG_Q_TIME
Tipo di dati: MQCFIL64
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

PutCount

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente nella coda, ad eccezione delle richieste MQPUT1 . Questo parametro è un elenco di numeri interi indicizzato dal valore di persistenza. Consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUTS
Tipo di dati: MQCFIL
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

PutFailCount

Descrizione: Il numero di tentativi non riusciti di inserire un messaggio nella coda

Identificativo: MQIAMO_PUTS_FAILED
Tipo di dati: MQCFIN
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

Put1Count

Descrizione: Il numero di messaggi persistenti e non persistenti inseriti correttamente nella coda utilizzando chiamate MQPUT1 . Questo parametro è un elenco di numeri interi indicizzato dal valore di persistenza. Consultare [Nota di riferimento 2](#).

Identificativo: MQIAMO_PUT1S
Tipo di dati: MQCFIL
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

Put1FailCount

Descrizione:	Il numero di tentativi non riusciti di inserire un messaggio utilizzando chiamate MQPUT1
Identificativo:	MQIAMO_PUT1S_FAILED
Tipo di dati:	MQCFIN
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

PutBytes

Descrizione:	Il numero di byte scritti nelle richieste di inserimento nella coda
Identificativo:	MQIAMO64_PUT_BYTES
Tipo di dati:	MQCFIL64
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

GetCount

Descrizione:	Il numero di richieste di ricezione distruttive riuscite per messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzato dal valore di persistenza. Consultare Nota di riferimento 2 .
Identificativo:	GET_MQIAMO
Tipo di dati:	MQCFIL
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

GetFailCount

Descrizione:	Il numero di richieste di ricezione distruttive non riuscite
Identificativo:	MQIAMO_GETS_NON RIUSCITO
Tipo di dati:	MQCFIN
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

GetBytes

Descrizione:	Il numero di byte letti nelle richieste di inserimento distruttive per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzato dal valore di persistenza. Consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_GET_BYTES
Tipo di dati:	MQCFIL64
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

BrowseCount

Descrizione:	Il numero di richieste di ricezione non distruttive riuscite per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzato dal valore di persistenza. Consultare Nota di riferimento 2 .
Identificativo:	MQIAMO_BROWSES
Tipo di dati:	MQCFIL
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

BrowseFailCount

Descrizione:	Il numero di richieste di ricezione non distruttive non riuscite
Identificativo:	MQIAMO_BROWSES_FAILED
Tipo di dati:	MQCFIN
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

BrowseBytes

Descrizione:	Il numero di byte letti nelle richieste di ricezione non distruttive per i messaggi persistenti e non persistenti. Questo parametro è un elenco di numeri interi indicizzato dal valore di persistenza. Consultare Nota di riferimento 2 .
Identificativo:	MQIAMO64_BROWSE_BYTES
Tipo di dati:	MQCFIL64
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

NonQueuedMsgCount

Descrizione:	Il numero di messaggi ignorati dalla coda e trasferiti direttamente a un'applicazione in attesa. Ignorare una coda può verificarsi solo in determinate circostanze. Questo numero rappresenta il numero di volte in cui IBM MQ è stato in grado di ignorare la coda e non il numero di volte in cui un'applicazione era in attesa.
Identificativo:	MQIAMO_MSGS_NOT_QUEUED
Tipo di dati:	MQCFIN
Incluso nel gruppo PCF:	<i>QStatisticsData</i>
Restituito:	Quando disponibile

ExpiredMsgCount

Descrizione:	Il numero di messaggi persistenti e non persistenti che sono stati eliminati perché scaduti prima che potessero essere richiamati.
Identificativo:	MQIAMO_MSGS_EXPIRED

Tipo di dati: MQCFIN
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

PurgeCount

Descrizione: Il numero di messaggi eliminati.
Identificativo: MQIAMO_MSGS_PURGED
Tipo di dati: MQCFIN
Incluso nel gruppo *QStatisticsData*
PCF:
Restituito: Quando disponibile

Dati del messaggio delle statistiche del canale

Utilizzare questa pagina per visualizzare la struttura di un messaggio di statistiche del canale

Nome messaggio: Messaggio statistiche canale.

Piattaforme: Tutti, tranne IBM MQ for z/OS.

Coda di sistema: SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Descrizione: Il nome del gestore code.
Identificativo: MQCA_Q_MGR_NAME.
Tipo di dati: MQCFST.
Lunghezza massima: MQ_Q_MGR_NAME_LENGTH.
Restituito: Sempre.

IntervalStartDate

Descrizione: La data all'inizio del periodo di controllo.
Identificativo: MQCAMO_START_DATE.
Tipo di dati: MQCFST.
Lunghezza massima: MQ_DATE_LENGTH.
Restituito: Sempre.

IntervalStartTime

Descrizione: L'ora all'inizio del periodo di controllo.
Identificativo: MQCAMO_START_TIME.
Tipo di dati: MQCFST.
Lunghezza massima: MQ_TIME_LENGTH.
Restituito: Sempre.

IntervalEndDate

Descrizione:	La data alla fine del periodo di monitoraggio
Identificativo:	DATA_END_CAMO.
Tipo di dati:	MQCFST.
Lunghezza massima:	MQ_DATE_LENGTH.
Restituito:	Sempre.

IntervalEndTime

Descrizione:	L'ora alla fine del periodo di monitoraggio
Identificativo:	MQCAMO_END_TIME.
Tipo di dati:	MQCFST.
Lunghezza massima:	MQ_ORA_LENGTH
Restituito:	Sempre.

CommandLevel

Descrizione:	Il livello di comando del gestore code.
Identificativo:	MQIA_COMMAND_LEVEL.
Tipo di dati:	MQCFIN.
Restituito:	Sempre.

ObjectCount

Descrizione:	Il numero di oggetti canale a cui è stato eseguito l'accesso nell'intervallo per cui sono stati registrati i dati statistici. Questo valore è impostato sul numero di gruppi PCF dati ChlStatistics contenuti nel messaggio.
Identificativo:	CONTEGGIO_OGGETTI_MQIAMO_
Tipo di dati:	MQCFIN.
Restituito:	Sempre.

ChlStatisticsData

Descrizione:	Parametri raggruppati che specificano i dettagli delle statistiche per un canale.
Identificativo:	MQGACF_CHL_STATISTICS_DATA.
Tipo di dati:	MQCFGR.

Parametri nel gruppo:

ChannelName
ChannelType
RemoteQmgr
ConnectionName
MsgCount
TotalBytes
NetTimeMin
NetTimeAvg
NetTimeMax
ExitTimeMin
ExitTimeAvg
ExitTimeMax
FullBatchCount
IncplBatchCount
AverageBatchSize
PutRetryCount

Restituito: Sempre.

ChannelName

Descrizione: Il nome del canale.
 Identificativo: MQCACH_CHANNEL_NAME.
 Tipo di dati: MQCFST.
 Lunghezza massima: LUNGHEZZA_NOME_CANALE_MQ.
 Restituito: Sempre.

ChannelType

Descrizione: Il tipo di canale.
 Identificativo: MQIACH_CHANNEL_TYPE.
 Tipo di dati: MQCFIN.
 Valori: I possibili valori sono:
MQCH_SENDER
 Canale mittente.
SERVER MQCHT
 Canale server.
MQCH_DESTINATARIO
 Canale ricevente.
RICHIESTA MQCHT_ER
 Canale richiedente.
CLUSRCVR MQCHT
 Canale ricevente cluster.
MQCHT_CLUSSDR
 Canale mittente cluster.

Restituito: Sempre.

RemoteQmgr

Descrizione:	Il nome del gestore code remoto.
Identificativo:	MQCA_REMOTE_Q_MGR_NAME.
Tipo di dati:	MQCFST.
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_
Restituito:	Quando disponibile.

ConnectionName

Descrizione:	Il nome connessione del gestore code remoto.
Identificativo:	MQCACH_CONNECTION_NAME.
Tipo di dati:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_CONN_MQ
Restituito:	Quando disponibile.

MsgCount

Descrizione:	Il numero di messaggi persistenti e non persistenti inviati o ricevuti.
Identificativo:	MQIAMO_MSGS.
Tipo di dati:	MQCFIN
Restituito:	Quando disponibile.

TotalBytes

Descrizione:	Il numero di byte inviati o ricevuti per i messaggi persistenti e non persistenti.
Identificativo:	MQIAMO64_BYTES.
Tipo di dati:	MQCFIN64.
Restituito:	Quando disponibile.

NetTimeMin

Descrizione:	Il round trip del canale registrato più breve misurato nell'intervallo di registrazione, in microsecondi.
Identificativo:	MQIAMO_NET_TIME_MIN.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

NetTimeAvg

Descrizione:	Il round trip medio del canale registrato misurato nell'intervallo di registrazione, in microsecondi.
Identificativo:	MQIAMO_NET_TIME_AVG.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

NetTimeMax

Descrizione:	Il round trip del canale registrato più lungo misurato nell'intervallo di registrazione, in microsecondi.
Identificativo:	MQIAMO_NET_TIME_MAX.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

ExitTimeMin

Descrizione:	Il tempo di registrazione più breve, in microsecondi, impiegato per l'esecuzione di un'uscita utente nell'intervallo di registrazione,
Identificativo:	MQIAMO_EXIT_TIME_MIN.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

ExitTimeAvg

Descrizione:	Il tempo medio registrato, in microsecondi, trascorso eseguendo un'uscita utente nell'intervallo di registrazione. Misurato in microsecondi.
Identificativo:	MQIAMO_EXIT_TIME_AVG.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

ExitTimeMax

Descrizione:	Il tempo più lungo registrato, in microsecondi, impiegato per l'esecuzione di un'uscita utente nell'intervallo di registrazione. Misurato in microsecondi.
Identificativo:	MQIAMO_EXIT_TIME_MAX.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

FullBatchCount

Descrizione:	Il numero di batch elaborati dal canale che sono stati inviati perché è stato raggiunto il valore degli attributi del canale BATCHSZ o BATCHLIM .
Identificativo:	MQIAMO_FULL_BATCHES.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

IncmplBatchCount

Descrizione:	Il numero di batch elaborati dal canale che sono stati inviati senza raggiungere il valore degli attributi del canale BATCHSZ o BATCHLIM .
Identificativo:	MQIAMO_INCOMPLETE_BATCHES.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

AverageBatchSize

Descrizione:	La dimensione batch media dei batch elaborati dal canale.
Identificativo:	MQIAMO_AVG_BATCH_SIZE.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

PutRetryCount

Descrizione:	Il numero di volte nell'intervallo di tempo in cui non è stato possibile inserire un messaggio ed è stato immesso un loop di tentativi.
Identificativo:	MQIAMO_PUT_RETRIES.
Tipo di dati:	MQCFIN.
Restituito:	Quando disponibile.

Note di riferimento

Utilizzare questa pagina per visualizzare le note a cui fanno riferimento le descrizioni della struttura dei messaggi di statistiche e account

Le seguenti descrizioni dei dati del messaggio fanno riferimento alle seguenti note:

- [“Dati del messaggio di account MQI” a pagina 160](#)
 - [“Dati del messaggio di account coda” a pagina 172](#)
 - [“Dati del messaggio delle statistiche MQI” a pagina 183](#)
 - [“Dati messaggio statistiche coda” a pagina 194](#)
 - [“Dati del messaggio delle statistiche del canale” a pagina 201](#)
1. Questo parametro è relativo agli oggetti IBM MQ . Questo parametro è un array di valori (MQCFIL o MQCFIL64) indicizzati dalle seguenti costanti:

Tipo oggetto	Contesto valore
MQOT_Q (1)	Contiene il valore relativo agli oggetti coda.
ELENCO NOMI MQOT_(2)	Contiene il valore relativo agli oggetti elenco nomi.
PROCESSO MQOT_(3)	Contiene il valore relativo agli oggetti processo.
Gestore code MQOT_(5)	Contiene il valore relativo agli oggetti gestore code.
CANALE_MQOT_(6)	Contiene il valore correlato agli oggetti canale.
MQOT_AUTH_INFO (7)	Contiene il valore relativo agli oggetti delle informazioni di autenticazione.
TOPIC MQOT_(8)	Contiene il valore relativo agli oggetti argomento.

Nota: Viene restituito un array di 13 valori MQCFIL o MQCFIL64 , ma solo quelli elencati sono significativi.

2. Questo parametro è relativo ai messaggi IBM MQ . Questo parametro è un array di valori (MQCFIL o MQCFIL64) indicizzati dalle seguenti costanti:

Tabella 25. Array indicizzato per valore di persistenza	
Costante	Valore
1	Contiene il valore per i messaggi non persistenti.
2	Contiene il valore per i messaggi persistenti.

Nota: L'indice per ciascuno di questi array inizia da zero, quindi un indice di 1 fa riferimento alla seconda riga dell'array. Gli elementi di questi array non elencati in queste tabelle non contengono informazioni su statistiche o account.

Traccia attività applicazione

La traccia dell'attività dell'applicazione produce informazioni dettagliate sul comportamento delle applicazioni connesse a un gestore code. Traccia il comportamento di una applicazione e fornisce una vista dettagliata dei parametri utilizzati da un'applicazione quando interagisce con risorse IBM MQ. Inoltre, mostra la sequenza di chiamate MQI emesse da una applicazione.


Utilizzare la traccia dell'attività dell'applicazione quando si richiedono più informazioni di quelle fornite dal monitoraggio eventi, dal monitoraggio messaggi, dai messaggi di statistiche e di account e dal monitoraggio in tempo reale.

Nota: La traccia delle attività viene generata sulla connessione IBM MQ per ogni applicazione; quindi, se i messaggi di account sono abilitati, le operazioni di traccia delle attività vengono conteggiate rispetto alle informazioni di account MQI per ogni applicazione.

IBM MQ supporta due metodi di raccolta dei dati di traccia delle applicazioni.

- Raccolta centrale delle informazioni di traccia dell'attività dell'applicazione, dove la traccia dell'attività dell'applicazione viene raccolta e letta scrivendo i messaggi PCF della traccia dell'attività sulla coda di sistema SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.
- Sottoscrizione ai dati di traccia dell'attività, scritti in argomenti speciali del sistema IBM MQ.

Tenere presente che la traccia attività non è supportata da IBM MQ for z/OS.

 Poiché ora è possibile specificare i nomi delle applicazioni nella maggior parte dei linguaggi di programmazione supportati da IBM MQ, fare riferimento a [Specifica del nome dell'applicazione nei linguaggi di programmazione supportati](#) per ulteriori informazioni.

Configurazione della raccolta centrale delle informazioni di traccia dell'attività dell'applicazione

Un messaggio di traccia dell'attività dell'applicazione è un messaggio PCF. Configurare la traccia attività utilizzando un file di configurazione. Per configurare la raccolta centrale delle informazioni di traccia dell'attività dell'applicazione, impostare l'attributo del gestore code ACTVTRC. È possibile sovrascrivere questa impostazione a livello di connessione utilizzando le opzioni MQCONNX o a livello di stanza dell'applicazione utilizzando il file di configurazione della traccia dell'attività.

Informazioni su questa attività

I messaggi di traccia dell'attività sono composti da una struttura MQMD: una struttura di intestazione PCF (MQCFH), seguita da un numero di parametri PCF. Una sequenza di gruppi PCF di dati ApplicationTracesegue i parametri PCF. Questi gruppi PCF raccolgono informazioni sulle operazioni MQI eseguite da una applicazione durante la connessione a un gestore code. Configurare la traccia dell'attività utilizzando un file di configurazione denominato mqat.ini.

Per controllare se vengono raccolte o meno le informazioni di traccia dell'attività dell'applicazione, configurare una o più delle seguenti impostazioni:

1. L'attributo del gestore code ACTVTRC.
2. Le impostazioni ACTVCONO (nella struttura MQCNO passata in MQCONNX).

3. La stanza corrispondente per l'applicazione nel file di configurazione della traccia attività `mqt.ini`.

La sequenza precedente è significativa. L'attributo `ACTVTRC` viene sovrascritto dalle impostazioni `ACTVCONO`, che vengono sovrascritte dalle impostazioni nel file `mqt.ini`.

Le voci di traccia vengono scritte dopo che ogni operazione è stata completata, se non diversamente specificato. Queste voci vengono scritte per la prima volta nella coda di sistema `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`, quindi scritto nei messaggi di traccia dell'attività dell'applicazione quando l'applicazione si disconnette dal gestore code. Per applicazioni di lunga durata, i messaggi intermedi vengono scritti se si verifica uno dei seguenti eventi:

- La durata della connessione raggiunge un valore di timeout definito.
- Il numero di operazioni raggiunge un numero specificato.
- La quantità di dati raccolti in memoria raggiunge la lunghezza massima del messaggio consentita per la coda.

Impostare il valore di timeout utilizzando il parametro **ActivityInterval**. Si imposta il numero di operazioni utilizzando il parametro **ActivityCount**. Entrambi i parametri vengono specificati nel file di configurazione della traccia attività `mqt.ini`.

L'abilitazione della traccia dell'attività dell'applicazione può influenzare le prestazioni. Il sovraccarico può essere ridotto ottimizzando le impostazioni **ActivityCount** e **ActivityInterval**. Consultare [“Ottimizzazione dell'impatto sulle prestazioni della traccia dell'attività dell'applicazione”](#) a pagina 215.

Il modo più semplice per visualizzare il contenuto dei messaggi di traccia dell'attività dell'applicazione consiste nell'utilizzare [“programma di esempio `amqsact`”](#) a pagina 216.

Procedura

1. [“Impostazione di `ACTVTRC` per controllare la raccolta delle informazioni di traccia dell'attività”](#) a pagina 208.
2. [“Impostazione delle opzioni `MQCONN` per controllare la raccolta delle informazioni di traccia dell'attività”](#) a pagina 209.
3. [“Configurazione del comportamento della traccia dell'attività utilizzando `mqt.ini`”](#) a pagina 210.
4. [“Ottimizzazione dell'impatto sulle prestazioni della traccia dell'attività dell'applicazione”](#) a pagina 215.

Impostazione di `ACTVTRC` per controllare la raccolta delle informazioni di traccia dell'attività

Utilizzare l'attributo del gestore code `ACTVTRC` per controllare la raccolta di informazioni di traccia dell'attività dell'applicazione `MQI`

Informazioni su questa attività

I messaggi di traccia dell'attività dell'applicazione vengono generati solo per le connessioni che iniziano dopo l'abilitazione della traccia dell'attività dell'applicazione. Il parametro **ACTVTRC** può avere i seguenti valori:

Attivo

La raccolta della traccia dell'attività API è abilitata

Disattivo

La raccolta della traccia dell'attività dell'API è disabilitata

Nota: L'impostazione **ACTVTRC** può essere sovrascritta dal parametro **ACTVCONO** del gestore code. Se si imposta il parametro **ACTVCONO** su `ENABLED`, l'impostazione **ACTVTRC** può essere sovrascritta per una determinata connessione utilizzando il campo **Options** nella struttura `MQCNO`. Consultare [“Impostazione delle opzioni `MQCONN` per controllare la raccolta delle informazioni di traccia dell'attività”](#) a pagina 209.

Esempio

Per modificare il valore del parametro **ACTVTRC**, utilizzare il comando MQSC ALTER QMGR. Ad esempio, per abilitare la raccolta di informazioni di traccia dell'attività dell'applicazione MQI, utilizzare il seguente comando MQSC:

```
ALTER QMGR ACTVTRC(ON)
```

Operazioni successive

Il modo più semplice per visualizzare il contenuto dei messaggi di traccia dell'attività dell'applicazione consiste nell'utilizzare [“programma di esempio amqsact”](#) a pagina 216.

L'abilitazione della traccia dell'attività dell'applicazione può influenzare le prestazioni. Il sovraccarico può essere ridotto ottimizzando le impostazioni **ActivityCount** e **ActivityInterval**. Consultare [“Ottimizzazione dell'impatto sulle prestazioni della traccia dell'attività dell'applicazione”](#) a pagina 215.

Impostazione delle opzioni MQCONNX per controllare la raccolta delle informazioni di traccia dell'attività

Se l'attributo del gestore code **ACTVCONO** è impostato su ENABLED, è possibile utilizzare il parametro **ConnectOpts** nella chiamata MQCONNX per abilitare o disabilitare i report di attività dell'applicazione in base alla connessione. Queste opzioni sovrascrivono il comportamento della traccia dell'attività definito dall'attributo del gestore code **ACTVTRC** e possono essere sovrascritte dalle impostazioni nel file di configurazione della traccia dell'attività `mqt.ini`.

Procedura

1. Impostare l'attributo del gestore code **ACTVCONO** su ENABLED.

Nota: Se un'applicazione tenta di modificare il comportamento di account di un'applicazione utilizzando il parametro **ConnectOpts**, e l'attributo QMGR **ACTVCONO** è impostato su DISABLED, non viene restituito alcun errore all'applicazione e la raccolta della traccia delle attività viene definita dagli attributi del gestore code o dal file di configurazione della traccia delle attività `mqt.ini`.

2. Impostare il parametro **ConnectOpts** nella chiamata MQCONNX su MQCNO_ACTIVITY_TRACE_ENABLED.

Il parametro **ConnectOpts** nella chiamata MQCONNX può avere i seguenti valori:

MQCNO_ACTIVITY_TRACE_DISABLED

La traccia attività è disabilitata per la connessione.

MQCNO_ACTIVITY_TRACE_ENABLED

La traccia attività è abilitata per la connessione.

Nota: Se un'applicazione seleziona MQCNO_ACTIVITY_TRACE_ENABLED e MQCNO_ACTIVITY_TRACE_DISABLED per MQCONNX, la chiamata ha esito negativo con un codice motivo MQRC_OPTIONS_ERROR.

3. Verificare che queste impostazioni della traccia attività non vengano sovrascritte dalle impostazioni nel file di configurazione della traccia attività `mqt.ini`.

Consultare [“Configurazione del comportamento della traccia dell'attività utilizzando mqt.ini”](#) a pagina 210.

Operazioni successive

Il modo più semplice per visualizzare il contenuto dei messaggi di traccia dell'attività dell'applicazione consiste nell'utilizzare [“programma di esempio amqsact”](#) a pagina 216.

L'abilitazione della traccia dell'attività dell'applicazione può influenzare le prestazioni. Il sovraccarico può essere ridotto ottimizzando le impostazioni **ActivityCount** e **ActivityInterval**. Consultare [“Ottimizzazione dell'impatto sulle prestazioni della traccia dell'attività dell'applicazione”](#) a pagina 215.

Configurazione del comportamento della traccia dell'attività utilizzando mqat.ini

Il comportamento della traccia attività viene configurato utilizzando un file di configurazione denominato mqat.ini. Questo file viene utilizzato per definire il livello e la frequenza dei dati di traccia delle attività di report. Il file fornisce anche un modo per definire le regole per abilitare e disabilitare la traccia dell'attività in base al nome di una applicazione.

Informazioni su questa attività

Linux **AIX** Su sistemi AIX and Linux , mqat.ini si trova nella directory dei dati del gestore code, che è la stessa ubicazione del file qm.ini .

Windows Sui sistemi Windows , mqat.ini si trova nella directory dei dati del gestore code C:\Program Files\IBM\WebSphere MQ\mqgrs\queue_manager_name. Gli utenti che eseguono le applicazioni da tracciare hanno bisogno dell'autorizzazione per leggere questo file.

Nota: I gestori code migrati da IBM WebSphere MQ 7.1 o versioni precedenti avranno il file mqat.ini mancante. In questi casi, il file mqat.ini deve essere creato manualmente e 660 autorizzazioni devono essere impostate sul file.

Quando il file mqat.ini viene modificato, le connessioni IBM MQ appena create verranno elaborate in base alla versione modificata. Le connessioni esistenti continueranno a utilizzare la versione precedente a meno che i parametri del gestore code non vengano modificati, ad esempio dopo un comando ALTER QMGR .

Questo file segue lo stesso formato della chiave stanza e della coppia parametro - valore dei file mqs.ini e qm.ini .

Il file è composto da una singola sezione, **AllActivityTrace**, per configurare il livello e la frequenza di notifica dei dati di traccia dell'attività per impostazione predefinita per tutta la traccia dell'attività.

Il file può contenere anche più stanze **ApplicationTrace** . Ognuno di questi, definisce una regola per il comportamento della traccia per una o più connessioni, in base alla corrispondenza del nome dell'applicazione delle connessioni alla regola.

Stanza di tracciaAllActivity

Una singola stanza **AllActivityTrace** definisce le impostazioni per la traccia dell'attività applicata a tutte le connessioni IBM MQ , a meno che non venga sovrascritta.

I singoli valori nella stanza **AllActivityTrace** possono essere sovrascritti da informazioni più specifiche in una stanza **ApplicationTrace** .

Se viene specificata più di una stanza **AllActivityTrace** , vengono utilizzati i valori nell'ultima stanza. I parametri mancanti dalla **AllActivity** selezionata assumono i valori predefiniti. I parametri e i valori delle stanze **AllActivityTrace** precedenti vengono ignorati.

I parametri seguenti possono essere specificati nella stanza **AllActivityTrace** :

Nome	Valori (valore predefinito in grassetto)	Descrizione
ActivityInterval	0-99999999 (1)	Intervallo di tempo approssimativo in secondi tra i messaggi di traccia. Tutte le attività eseguite da una connessione in tale intervallo verranno scritte in un singolo messaggio. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando viene raggiunto il conteggio attività).

Tabella 26. Coppie parametro / valore che possono essere utilizzate nel file di configurazione di traccia attività (Continua)

Nome	Valori (valore predefinito in grassetto)	Descrizione
ActivityCount	0 - 99999999 (100)	Numero di operazioni MQI o XA tra i messaggi di traccia. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando è trascorso l'intervallo di attività).
TraceLevel	LOW / MEDIUM / ALTO	Quantità di dettagli del parametro tracciati per ciascuna operazione. La descrizione delle singole operazioni descrive i parametri inclusi per ciascun livello di traccia.
Dati TraceMessage	0 - 104 857 600 (massimo 100 MB)	La quantità di dati del messaggio tracciati in byte per le operazioni MQGET, MQPUT, MQPUT1e Callback
Messaggio StopOnGetTrace	ON / OFF	L'utilizzo della traccia dell'attività, per tracciare le applicazioni che stanno elaborando anche i messaggi di traccia dell'attività, non è consigliabile a causa di un possibile loop.
SubscriptionDelivery	BATCH / IMMEDIATE	Determina se i parametri ActivityInterval e ActivityCount devono essere utilizzati quando sono presenti una o più sottoscrizioni di traccia dell'attività. L'impostazione di questo parametro su IMMEDIATE determina la sovrascrittura dei valori di ActivityInterval e ActivityCount con i valori effettivi di 1 quando i dati di traccia hanno una sottoscrizione corrispondente. Ogni record di traccia dell'attività non viene raggruppato in batch con altri record della stessa connessione e viene invece consegnato alla sottoscrizione immediatamente senza alcun ritardo.

Stanza ApplicationTrace

Una stanza ApplicationTrace contiene una regola che definisce quali connessioni IBM MQ verranno tracciate o meno in base al nome dell'applicazione. Facoltativamente, il comportamento predefinito definito in Allsettings che sovrascrive il livello di traccia globale e le impostazioni di frequenza.




Questa stanza può includere i parametri ApplName, ApplFunction e ApplClass che vengono utilizzati in base alle regole di corrispondenza definite in Connection Matching Rules per stabilire se la stanza si applica o meno a una particolare connessione.

La stanza deve includere il parametro di traccia per determinare se questa regola attiva o disattiva la traccia dell'attività per le connessioni corrispondenti.

Una regola di disattivazione può essere utilizzata per disabilitare esplicitamente la traccia per nomi di applicazioni più specifici e per sovrascrivere l'impostazione ACTVTRC delle opzioni di connessione del gestore code o della traccia di attività.

I parametri seguenti possono essere specificati nella stanza **ApplicationTrace** :

Tabella 27. Coppie parametro / valore che possono essere utilizzate nel file di configurazione della traccia dell'applicazione

Nome	Valori (valore predefinito in grassetto)	Descrizione
Traccia	ON / OFF (Parametro obbligatorio - nessun valore predefinito)	Commutazione traccia attività. Questo switch può essere utilizzato nella sezione specifica dell'applicazione per determinare se la traccia dell'attività è attiva per l'ambito della stanza dell'applicazione corrente. Notare che questo valore sovrascrive le impostazioni ACTVTRC e ACTVCONO per il gestore code.
ApplName	Stringa di caratteri (parametro obbligatorio - nessun valore predefinito)	<p>Questo valore viene utilizzato per stabilire a quali applicazioni si applica la sezione ApplicationTrace . Viene associato al valore ApplName dalla struttura del contesto dell'uscita API (equivalente a MQMD.PutApplName). Il contenuto del valore ApplName varia a seconda dell'ambiente dell'applicazione.</p> <p>Per piattaforme diverse da z/OS, solo la porzione del nome file di MQAXC.ApplName corrisponde al valore nella sezione. I caratteri a sinistra del separatore di percorso più a destra vengono ignorati quando viene effettuato il confronto.</p> <p> Per applicazioni z/OS , l'intero MQAXC MQAXC.ApplName corrisponde al valore nella sezione.</p> <p>Un singolo carattere jolly (*) può essere utilizzato alla fine del valore ApplName per corrispondere a qualsiasi numero di caratteri dopo tale punto. Se il valore ApplName è impostato su un singolo carattere jolly (*), il valore ApplName corrisponde a tutte le applicazioni.</p>
  ApplFunction	Stringa di caratteri (valore predefinito *)	<p>Questo valore viene utilizzato per qualificare i programmi di applicazione a cui si applica la stanza ApplicationTrace e il valore App1Name .</p> <p>La stanza è facoltativa, ma è valida solo per i gestori code IBM i . Un singolo carattere jolly (*) può essere utilizzato alla fine del valore App1Name per corrispondere a qualsiasi numero di caratteri.</p> <p>Ad esempio, una stanza ApplicationTrace che specifica App1Name = * e App1Function = AMQSPUTO si applica a tutte le chiamate del programma AMQSPUTO da qualsiasi lavoro.</p>
ApplClass	USER / MCA / ALL	La classe dell'applicazione. Consultare la seguente tabella per una spiegazione di come i valori AppType corrispondono alle connessioni IBM MQ .

La seguente tabella mostra in che modo i valori di *AppClass* corrispondono ai campi **APICallerType** e **APIEnvironment** nella struttura del contesto di uscita API della connessione.

Tabella 28. I valori <i>Appclass</i> e come corrispondono ai campi <i>APICallerType</i> e <i>APIEnvironment</i>			
APPLCLASS	Tipo di chiamante API:	Ambiente API:	Descrizione
UTENTE	MQXACT_EXTERNAL	MQXE_ALTRO	Vengono tracciate solo le applicazioni utente
MCA	(Qualsiasi valore)	MQXE_MCA MQXE_MCA_CLNT MQXE_MCA_SVRCONN	Client e canali (amqrmppa)
TUTTO	(Qualsiasi valore)	(Qualsiasi valore)	Tutte le connessioni vengono tracciate



Attenzione: È necessario utilizzare un **APPLCLASS MCA** per le applicazioni utente del client, poiché una classe di *USER* non corrisponde a tali applicazioni.

Ad esempio, per tracciare l'applicazione di esempio **amqspu_tc**, è possibile utilizzare il seguente codice:

```

ApplicationTrace:
  ApplClass=MCA                                # Application type
                                              # Values: (USER | MCA | INTERNAL | ALL)
                                              # Default: USER
  ApplName=amqsputc    # Application name (may be wildcarded)
                                              # (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

```

Il `mqat.ini` predefinito generato quando viene creato un gestore code, contiene una singola regola per disabilitare esplicitamente la traccia attività per l'esempio di traccia attività fornito, **amqsact**.

Regole di corrispondenza connessione

Il gestore code applica le seguenti regole per determinare le impostazioni delle stanze da utilizzare per una connessione.

1. Un valore specificato nella stanza **AllActivityTrace** viene utilizzato per la connessione a meno che il valore non si verifichi anche in una stanza **ApplicationTrace** e la stanza soddisfi i criteri di corrispondenza per la connessione descritti nei punti [2](#), [3](#) e [4](#).
2. Il **App1Class** viene confrontato con il tipo di connessione IBM MQ. Se **App1Class** non corrisponde al tipo di connessione, la stanza viene ignorata per questa connessione.
3. Il valore *App1Name* nella stanza viene confrontato con la parte del nome file del campo **App1Name** dalla struttura del contesto di uscita API (MQAXC) per la connessione.

La parte del nome file deriva dai caratteri alla destra del carattere separatore di percorso finale (/ o \). Se la stanza **App1Name** include un carattere jolly (*), solo i caratteri alla sinistra del carattere jolly vengono confrontati con il numero equivalente di caratteri dal **App1Name** della connessione.

Ad esempio, se viene specificato un valore di stanza "FRE*", nel confronto vengono utilizzati solo i primi tre caratteri, in modo che "path/FREEDOM" e "path\FREDDY" corrispondano, ma "path/FRIEND" no. Se il valore *AppName* della stanza non corrisponde alla connessione **AppName**, la stanza viene ignorata per questa connessione.

4. Se più di una stanza corrisponde a **AppName** e **AppClass** della connessione, viene utilizzata la stanza con il **AppName** più specifico.

Il **AppName** più specifico è definito come quello che utilizza il maggior numero di caratteri per corrispondere al **AppName** della connessione.

Ad esempio, se il file *ini* contiene una stanza con **AppName** = "FRE*" e un'altra stanza con **AppName** = "FREE*", la stanza con **AppName** = "FREE*" viene scelta come la migliore corrispondenza per una connessione con **AppName** = "path/FREEDOM" perché corrisponde a quattro caratteri (mentre **AppName** = "FRE*" ne corrisponde solo tre).

5. Se, dopo aver applicato le regole nei punti 2, 3 e 4, esiste più di una stanza che corrisponde alle connessioni **AppName** e **AppClass** della connessione, verranno utilizzati i valori dell'ultima corrispondenza e tutte le altre stanze verranno ignorate.

Sostituzione delle impostazioni predefinite per ogni regola

Facoltativamente, il livello di traccia globale e le impostazioni di frequenza nella stanza **AllActivityTrace** possono essere sovrascritti per quelle connessioni che corrispondono a una stanza **ApplicationTrace**.

I parametri seguenti possono essere impostati in una stanza **ApplicationTrace**. Se non sono impostati, il valore viene ereditato dalle impostazioni della stanza **AllActivityTrace**:

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**
- **StopOnTraceMsg**

mqat.ini sintassi

Le regole di sintassi per il formato del file *mqat.ini* sono:

- Il testo che inizia con un hash o un punto e virgola è considerato un commento che si estende fino alla fine della riga.
- La prima riga significativa (non di commento) deve essere una chiave stanza.
- Una chiave di stanza consiste nel nome della stanza seguito da due punti.
- Una coppia parametro - valore è costituita dal nome di un parametro seguito da un segno di uguale e quindi dal valore.
- Solo una singola coppia parametro - valore può apparire su una linea. (Un valore di parametro non deve andare a capo su un'altra riga).
- Gli spazi iniziali e finali vengono ignorati. Non esiste alcun limite alla quantità di spazi vuoti tra i nomi delle stanze, i nomi dei parametri e i valori o le coppie parametro / valore. Le interruzioni di riga sono significative e non ignorate.
- La lunghezza massima per ogni riga è di 2048 caratteri.
- Le chiavi della stanza, i nomi dei parametri e i valori dei parametri delle costanti non sono sensibili al maiuscolo / minuscolo, ma i valori dei parametri della variabile (*AppName* e *DebugPath*) sono sensibili al maiuscolo / minuscolo.

Esempio di file di traccia attività dell'applicazione

Il seguente esempio mostra come vengono specificati i dati di configurazione nel file *ini* di traccia attività.

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
AppIName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
AppIName=amqsact*
Trace=OFF
```

La precedente stanza **AllActivityTrace** definisce il modo in cui la traccia dell'attività verrà eseguita per impostazione predefinita quando abilitata, tramite le regole ApplicationTrace o tramite l'attributo ACTVTRC del gestore code o in modo programmatico abilitato da un'applicazione.

La prima stanza **ApplicationTrace** definisce una regola che risulterà in qualsiasi attività MQI da parte di un'applicazione il cui nome inizia con "amqs" che viene tracciato. La traccia generata per queste applicazioni sarà molto dettagliata e includerà fino a 1000 byte di dati del messaggio. I parametri di intervallo e conteggio dell'attività verranno ereditati

La seconda stanza **ApplicationTrace** definisce una regola che disattiva la funzione di traccia per le applicazioni con nomi che iniziano con "amqsact" (l'esempio di traccia attività). Questa regola sovrascriverà la regola 'on' precedente per l'applicazione amqsact, senza alcuna traccia per tale applicazione.

Un esempio viene fornito anche come esempio denominato mqat . ini nella directory degli esempi C (la stessa directory del file amqsact.c). Questo file può essere copiato nella directory dei dati del gestore code, per i gestori code che sono stati migrati da una release precedente di IBM MQ.

Operazioni successive

L'abilitazione della traccia dell'attività dell'applicazione può influenzare le prestazioni. Il sovraccarico può essere ridotto ottimizzando le impostazioni **ActivityCount** e **ActivityInterval** . Consultare [“Ottimizzazione dell'impatto sulle prestazioni della traccia dell'attività dell'applicazione”](#) a pagina 215.

Ottimizzazione dell'impatto sulle prestazioni della traccia dell'attività dell'applicazione

L'abilitazione della traccia dell'attività dell'applicazione può comportare una penalizzazione delle prestazioni. Ciò può essere ridotto solo tracciando le applicazioni necessarie, aumentando il numero di applicazioni che svuotano la coda e ottimizzando **ActivityInterval**, **ActivityCount** e **TraceLevel** in mqat . ini.

Informazioni su questa attività

L'abilitazione della traccia dell'attività dell'applicazione in modo selettivo per un'applicazione o per tutte le applicazioni del gestore code può risultare in un'attività di messaggistica aggiuntiva e nel gestore code che richiede ulteriore spazio di memoria. In ambienti in cui le prestazioni di messaggistica sono critiche, ad esempio, in applicazioni con un carico di lavoro elevato o in cui uno SLA (Service Level Agreement) richiede un tempo di risposta minimo dal provider di messaggistica, potrebbe non essere appropriato raccogliere la traccia dell'attività dell'applicazione o potrebbe essere necessario modificare i dettagli o la frequenza dei messaggi di attività di traccia prodotti. I valori preimpostati di **ActivityInterval**, **ActivityCount** e **TraceLevel** nel file mqat . ini forniscono un bilanciamento predefinito di dettaglio e prestazioni. Tuttavia, è possibile regolare questi valori per soddisfare i requisiti di prestazioni e funzionali del sistema.

Procedura

- Tenere traccia solo delle applicazioni necessarie.

Eeguire questa operazione creando una stanza specifica dell'applicazione ApplicationTrace in `mqt.ini` modificando l'applicazione in modo da specificare `MQCNO_ACTIVITY_TRACE_ENABLED` nel campo delle opzioni nella struttura **MQCNO** di una chiamata MQCONN. Consultare [“Configurazione del comportamento della traccia dell'attività utilizzando mqt.ini”](#) a pagina 210 e [“Impostazione delle opzioni MQCONN per controllare la raccolta delle informazioni di traccia dell'attività”](#) a pagina 209.

- Prima di avviare la traccia, verificare che almeno un'applicazione sia in esecuzione e che sia pronta a recuperare i dati del messaggio di traccia attività da `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`.
- Mantenere la profondità della coda il più bassa possibile, aumentando il numero di applicazioni che svuotano la coda.
- Impostare il valore **TraceLevel** nel file `mqt.ini` per raccogliere la quantità minima di dati richiesti.

`TraceLevel=LOW` ha l'impatto più basso sulle prestazioni di messaggistica.

Consultare [“Configurazione del comportamento della traccia dell'attività utilizzando mqt.ini”](#) a pagina 210.

- Ottimizzare i valori **ActivityCount** e **ActivityInterval** in `mqt.ini` per regolare la frequenza con cui vengono generati i messaggi di traccia dell'attività.

Se si sta eseguendo la traccia di più applicazioni, i messaggi di traccia dell'attività potrebbero essere prodotti più velocemente di quanto non possano essere rimossi da `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`. Tuttavia, quando si riduce la frequenza con cui vengono generati i messaggi di traccia dell'attività, si aumenta anche lo spazio di memoria richiesto dal gestore code e la dimensione dei messaggi quando vengono scritti nella coda.

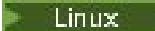


Operazioni successive

Il modo più semplice per visualizzare il contenuto dei messaggi di traccia dell'attività dell'applicazione consiste nell'utilizzare [“programma di esempio amqsact”](#) a pagina 216.

programma di esempio amqsact

amqsact formatta i messaggi di Application Activity Trace e viene fornito con IBM MQ.

Il programma compilato si trova nella directory degli esempi:

-   Su AIX and Linux `MQ_INSTALLATION_PATH/samp/bin`
-  Su Windows `MQ_INSTALLATION_PATH\tools\c\Samples\Bin`

Modalità di visualizzazione

Per impostazione predefinita, **amqsact** in modalità di visualizzazione elabora i messaggi su `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`. È possibile sovrascrivere questo comportamento specificando un nome coda o una stringa di argomenti.

È inoltre possibile controllare il periodo di traccia visualizzato e specificare se i messaggi di traccia dell'attività vengono rimossi o conservati dopo la visualizzazione.

Output di esempio per la modalità dinamica

Il seguente comando genera e visualizza i messaggi di traccia dell'attività per tutte le connessioni effettuate dalle applicazioni che iniziano con il testo "amqs". Dopo 30 secondi di inattività, il programma **amqsact** termina e non vengono generati nuovi dati di traccia dell'attività.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

Il seguente comando genera e visualizza i messaggi di traccia dell'attività per qualsiasi attività su QMGR1.TO.QMGR2 . Dopo 10 secondi di inattività, il programma **amqsact** termina e non vengono generati nuovi dati di traccia dell'attività.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

Il seguente comando genera e visualizza messaggi di traccia dell'attività dettagliati per qualsiasi attività sulla connessione IBM MQ esistente che ha un CONN di "6B576B5420000701" e un EXTCONN di "414D5143514D47523120202020202020". Dopo un minuto di inattività, il programma **amqsact** termina e non viene generato alcun nuovo dato di traccia dell'attività.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

Sottoscrizione alle informazioni di traccia dell'attività dell'applicazione

È possibile sottoscrivere in modo dinamico le informazioni di traccia dell'attività dell'applicazione come alternativa alla raccolta delle informazioni tramite la configurazione del gestore code.

Informazioni su questa attività

La traccia dell'attività dell'applicazione traccia il comportamento di un'applicazione e fornisce una vista dettagliata dei parametri utilizzati da un'applicazione quando interagisce con risorse IBM MQ . Inoltre, mostra la sequenza di chiamate MQI emesse da una applicazione.

Oltre alla scrittura dei dati di traccia nella coda di sistema, il prodotto fornisce la possibilità di sottoscrivere dinamicamente i dati di traccia dell'attività, scritti in argomenti speciali del sistema IBM MQ , piuttosto che raccogliere le informazioni tramite la configurazione del livello del gestore code.

Tenere presente che da IBM MQ 9.0 il prodotto non utilizza uscite per questo scopo. Se sono state precedentemente utilizzate delle uscite per tracciare l'attività dell'applicazione, è necessario passare all'utilizzo del metodo di sostituzione per raccogliere la traccia dell'attività dell'applicazione.

La creazione di una sottoscrizione abilita la traccia dell'attività. Non è necessario impostare gli attributi del gestore code o dell'applicazione come per la raccolta centrale dei dati di traccia. Tuttavia, qualsiasi blocco esplicito della traccia dell'attività disabilitando la traccia a livello del gestore code o dell'applicazione blocca anche la consegna della traccia dell'attività a tutte le sottoscrizioni corrispondenti.

Procedura

- [“Sottoscrizioni alla traccia dell'attività dell'applicazione” a pagina 219](#)
- [“Creazione di sottoscrizioni alla traccia dell'attività dell'applicazione” a pagina 220](#)
- [“Utilizzo di amqsact per visualizzare i messaggi di traccia” a pagina 222](#)
- [“Configurazione dei livelli traccia utilizzando mqat.ini” a pagina 224](#)

Sottoscrizioni alla traccia dell'attività dell'applicazione

È possibile sottoscrivere un argomento del sistema IBM MQ per raccogliere le informazioni di traccia dell'attività dell'applicazione.

Sottoscrivere una stringa di argomenti di sistema IBM MQ speciale che rappresenta l'attività da tracciare. La sottoscrizione genera automaticamente i messaggi di dati di traccia dell'attività e li pubblica nella coda di destinazione della sottoscrizione. Se si elimina la sottoscrizione, la generazione dei dati di traccia dell'attività si arresta per tale sottoscrizione.

Una sottoscrizione può tracciare l'attività su una delle seguenti risorse:

- Un'applicazione specificata
- Un canale IBM MQ specificato
- Una connessione IBM MQ esistente

È possibile creare più sottoscrizioni, con stringhe di argomenti differenti o uguali. Quando si creano più sottoscrizioni con le stesse stringhe di argomenti di traccia dell'attività di sistema, ciascuna sottoscrizione riceve una copia dei dati di traccia dell'attività e ciò potrebbe avere implicazioni negative sulle prestazioni.

L'abilitazione di qualsiasi livello di traccia dell'attività potrebbe avere effetti negativi sulle prestazioni. Maggiore è il numero di sottoscrizioni o di risorse sottoscritte, maggiore è il potenziale sovraccarico delle prestazioni. Per ridurre il sovraccarico della raccolta della traccia dell'attività, i dati vengono scritti nei messaggi e consegnati alle sottoscrizioni in modo asincrono dall'attività dell'applicazione stessa. Spesso, più operazioni vengono scritte in un singolo messaggio di dati di traccia dell'attività. L'operazione asincrona può introdurre un ritardo tra l'operazione dell'applicazione e la ricezione dei dati di traccia che registrano l'operazione.

Creazione di sottoscrizioni alla traccia dell'attività dell'applicazione

È possibile creare sottoscrizioni a argomenti specifici per raccogliere i dati di traccia dell'attività dell'applicazione.

Quando una sottoscrizione viene creata rispetto a specifiche stringhe di argomenti di sistema, i messaggi di dati PCF di traccia dell'attività appropriati vengono pubblicati automaticamente su tale sottoscrizione. Per informazioni dettagliate sulla sottoscrizione agli argomenti, vedere [Messaggistica di pubblicazione / sottoscrizione](#).

Le stringhe argomento hanno il formato:

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifier
```

dove:

- *qmgr_name* specifica il gestore code a cui è connessa l'applicazione tracciata. *qmgr_name* è il nome del gestore code con tutti i caratteri vuoti finali rimossi e tutti i caratteri barra (/) sostituiti da una e commerciale (&) carattere.
- *tipo_risorsa* specifica il tipo di dati della risorsa per cui viene effettuata la raccolta ed è una delle seguenti stringhe:
 - App1Name per specificare un'applicazione. La richiesta sottoscrive tutte le connessioni IBM MQ che hanno un nome applicazione che corrisponde a quello specificato da *resource_identifier*.
 - ChannelName per specificare un canale IBM MQ .
 - ConnectionId per specificare una connessione IBM MQ .
- *resource_identifier* identifica la risorsa effettiva. Il formato dipende dal tipo di risorsa:
 - Per un tipo di risorsa di App1Name, *resource_identifier* è la parte finale (il valore che segue l'ultimo / o \) del nome dell'applicazione come visualizzato dal gestore code, con gli eventuali caratteri vuoti finali rimossi. Il valore corrisponde al valore App1Name dalla struttura del contesto di uscita API (MQAXC). Il App1Name di una connessione viene restituito come valore APPLTAG quando si utilizza il comando MQSC **DISPLAY CONN**.
 - Per un tipo di risorsa ChannelName, *resource_identifier* è il nome del canale di cui eseguire la traccia. Se il nome canale identifica un canale SVRCONN, viene eseguita la traccia di tutta l'attività dell'applicazione per i client connessi. Se il nome del canale identifica un gestore code nel canale

del gestore code, vengono tracciati i messaggi in entrata e in uscita. *resource_identifier* è il nome del canale con tutti i caratteri vuoti finali rimossi e tutti i caratteri '/' sostituiti da ' & ' carattere.

- Per un tipo di risorsa *ConnectionId*, *resource_identifier* è l'identificativo di connessione univoco assegnato a ciascuna connessione. L'identificativo di collegamento nella stringa argomento è il valore completo di 24 byte scritto come stringa esadecimale. Questo valore è la concatenazione di EXTCONN seguito dai valori CONN restituiti dal comando MQSC **DISPLAY CONN**.

Puoi utilizzare i caratteri jolly in un *resource_identifier* per mettere in corrispondenza più identità di risorsa in una singola sottoscrizione. Il carattere jolly può essere nello stile dell'argomento predefinito ('#' o '+') o nello stile del carattere ('*' o '?'). Quando si utilizza il carattere jolly dello stile argomento, non può essere combinato con parte di un nome risorsa, può essere utilizzato solo per corrispondere a tutte le applicazioni, i canali o le connessioni possibili. L'utilizzo di qualsiasi carattere jolly aumenta il livello dei dati di traccia generati, il che può influire sulle prestazioni.

Per sottoscrivere queste stringhe di argomenti, è necessario disporre dell'autorizzazione di "sottoscrizione". Gli argomenti di sistema non ereditano autorizzazioni dalla root della struttura ad albero degli argomenti del gestore code. A un utente deve essere concesso l'accesso a un oggetto argomento gestito al punto \$SYS/MQ nella struttura ad albero degli argomenti. È possibile effettuare la sottoscrizione se si dispone dell'accesso al SISTEMA SYSTEM.ADMIN.TOPIC, sebbene questo conceda l'accesso a tutte le stringhe di argomenti \$SYS/MQ, non solo alla traccia dell'attività. Per controllare l'accesso in modo più specifico, è possibile definire nuovi oggetti argomento gestiti per punti più profondi nella struttura ad albero, per tutta la traccia dell'attività o, ad esempio, per un nome di applicazione o di canale specifico.

Esempi

L'esempio riportato di seguito mostra una stringa di argomento per un'applicazione denominata amqspu^t in esecuzione su un sistema Windows:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqsput.exe
```

Il seguente esempio mostra una stringa di argomenti per un canale:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

Il seguente esempio mostra una stringa di argomenti per una connessione:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ConnectionId/  
414D5143514D475231202020202020206B576B542000701
```

Il seguente esempio mostra una stringa di argomenti che crea una sottoscrizione per tracciare i dati per tutti i canali sul gestore code QMGR1:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

Il seguente esempio mostra una stringa di argomenti che crea una sottoscrizione per tracciare i dati per le applicazioni i cui nomi iniziano con "amq" (notare che per utilizzare il carattere jolly "*", la sottoscrizione deve essere creata utilizzando il modello carattere jolly):

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amq*
```

Concetti correlati

[“Argomenti di sistema per il monitoraggio e la traccia dell'attività” a pagina 296](#)

Gli argomenti di sistema nelle strutture ad albero degli argomenti del gestore code vengono utilizzati per il controllo delle risorse (alcuni dei quali sono simili al contenuto dei messaggi delle statistiche) e come un modo per utilizzare la traccia dell'attività dell'applicazione.

Utilizzo di *amqsact* per visualizzare i messaggi di traccia

È possibile utilizzare il **amqsact** programma per generare e visualizzare i messaggi di traccia.

Il programma **amqsact** è un esempio IBM MQ . Per utilizzare questo esempio, è necessario utilizzare il file eseguibile connesso al client, **amqsactc**. Il file eseguibile si trova nella directory degli esempi:

- Su piattaforme Linux e UNIX, *MQ_INSTALLATION_PATH/samp/bin64*
- Su piattaforme Windows, *MQ_INSTALLATION_PATH\tools\c\Samples\Bin64*

È possibile utilizzare **amqsact** in due modi:

Modalità di visualizzazione

Formattare e visualizzare i messaggi di dati di traccia dell'attività che vengono consegnati a SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

Modalità dinamica

Creare una sottoscrizione a una serie di risorse e visualizzare la traccia attività generata eseguendo **amqsact**.

Modalità di visualizzazione

Per impostazione predefinita, **amqsact** in modalità di visualizzazione elabora i messaggi su SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. È possibile sovrascrivere questo comportamento specificando un nome coda o una stringa di argomenti. La traccia dell'attività deve essere abilitata utilizzando uno dei metodi descritti in [Raccolta delle informazioni di traccia dell'attività dell'applicazione](#). È possibile controllare il periodo di traccia visualizzato e specificare se i messaggi di traccia dell'attività vengono rimossi o conservati dopo la visualizzazione. In modalità di visualizzazione, **amqsact** utilizza i seguenti argomenti:

-m nome_gestore_coda

Obbligatorio. Specificare il gestore code per cui vengono raccolti i messaggi di traccia.

-q nome_coda

Visualizza solo i messaggi di traccia relativi alla coda denominata.

-t stringa_argomento

Visualizzare solo i messaggi di traccia correlati all'argomento specificato.

-b

Specificare che i messaggi di traccia vengono conservati dopo la visualizzazione.

-v

Visualizzare i messaggi di traccia in modalità dettagliata.

-d profondità

Il numero di messaggi da visualizzare.

-w timeout

Specificare un timeout. Se non viene visualizzato alcun messaggio di traccia in quel periodo, **amqsact** esce.

-s ora_inizio

Utilizzare questo argomento con l'argomento **-e** per specificare un periodo di tempo. Vengono visualizzati i messaggi di traccia dal periodo di tempo specificato.

-e ora_fine

Utilizzare questo argomento con l'argomento **-s** per specificare un periodo di tempo. Vengono visualizzati i messaggi di traccia dal periodo di tempo specificato.

Ad esempio, il seguente comando visualizza i messaggi di traccia dell'attività che si trovano su SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE e elimina i messaggi dopo la visualizzazione:

```
amqsact -m QMGR1
```

Il seguente comando visualizza i messaggi di traccia attività sulla coda specificata, SUB.QUEUE e cancella i messaggi dopo la visualizzazione. I messaggi continuano ad essere visualizzati fino a quando non

trascorre un periodo di 30 secondi senza nuovi messaggi. Questo comando, ad esempio, può essere utilizzato con una sottoscrizione ad una stringa di argomenti del sistema di traccia attività.

```
amqact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

Il seguente comando visualizza in formato verbose tutti i dati di traccia dell'attività attualmente conservati sul SISTEMA SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE che si è verificato nel periodo di 20 minuti specificato. I messaggi rimarranno nella coda dopo la visualizzazione.

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

Modalità dinamica

Abilitare la modalità dinamica specificando un nome applicazione, un nome canale o un identificativo di connessione come argomento per **amqsact**. È possibile utilizzare caratteri jolly nel nome. In modalità dinamica, i dati di traccia dell'attività vengono abilitati all'inizio dell'esempio utilizzando una sottoscrizione non durevole a un argomento di sistema. La raccolta dei dati di traccia dell'attività si arresta quando **amqsact** si arresta. È necessario specificare un timeout per **amqsact** in modalità dinamica. È possibile eseguire più copie di **amqsact** contemporaneamente e ciascuna istanza riceve una copia di tutti i dati di traccia delle attività. In modalità dinamica, **amqsact** utilizza i seguenti argomenti:

-m nome_gestore_coda

Obbligatorio. Specificare il gestore code per cui vengono raccolti i messaggi di traccia.

-w timeout

Obbligatorio. Specificare un timeout. Se non viene visualizzato alcun messaggio di traccia in quel periodo, **amqsact** esce.

-a nome_applicazione

Specificare un'applicazione per cui raccogliere i messaggi.

-c nome_canale

Specificare un canale per cui raccogliere i messaggi.

-i ID_connessione

Specificare una connessione per cui raccogliere i messaggi.

-v

Visualizzare i messaggi di traccia in modalità dettagliata.

Ad esempio, il seguente comando genera e visualizza i messaggi di traccia dell'attività per tutte le connessioni effettuate dalle applicazioni denominate "amqsget.exe". Dopo 30 secondi di inattività, il programma **amqsact** termina e non vengono generati nuovi dati di traccia dell'attività.

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

Il seguente comando genera e visualizza i messaggi di traccia dell'attività per tutte le connessioni effettuate dalle applicazioni che iniziano con il testo "amqs". Dopo 30 secondi di inattività, il programma **amqsact** termina e non vengono generati nuovi dati di traccia dell'attività.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

Il seguente comando genera e visualizza i messaggi di traccia dell'attività per qualsiasi attività su QMGR1.TO.QMGR2 . Dopo 10 secondi di inattività, il programma **amqsact** termina e non vengono generati nuovi dati di traccia dell'attività.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

Il seguente comando genera e visualizza i messaggi di traccia dell'attività per qualsiasi attività su qualsiasi canale. Dopo 10 secondi di inattività, il programma **amqsact** termina e non vengono generati nuovi dati di traccia dell'attività.

```
amqsactc -m QMGR1 -w 10 -c #
```

Il seguente comando genera e visualizza messaggi di traccia dell'attività dettagliati per qualsiasi attività sulla connessione IBM MQ esistente che ha un CONN di "6B576B5420000701" e un EXTCNN di "414D5143514D47523120202020202020". Dopo un minuto di inattività, il programma **amqsact** termina e non viene generato alcun nuovo dato di traccia dell'attività.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

Configurazione dei livelli traccia utilizzando mqat.ini

Configurare i livelli di traccia per un gestore code impostando i valori per la stanza di traccia AllActivity del file di configurazione mqat.ini.

È possibile impostare i seguenti valori per la stanza AllActivityTrace:

ActivityInterval

Intervallo di tempo in secondi tra i messaggi di traccia. La traccia attività non utilizza un thread timer, quindi il messaggio di traccia non viene scritto nel momento esatto in cui il tempo trascorre, ma viene scritto quando viene eseguita la prima operazione MQI una volta trascorso l'intervallo di tempo. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando viene raggiunto il conteggio attività). L'impostazione predefinita è 1.

ActivityCount

Numero di operazioni MQI tra i messaggi di traccia. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando trascorre l'intervallo di attività). Il valore predefinito è 100.

TraceLevel

Quantità di dettagli del parametro tracciati per ogni operazione. La descrizione delle singole operazioni descrive i parametri inclusi per ciascun livello di traccia. Impostare su LOW, MEDIUM o HIGH. Il valore predefinito è MEDIUM.

Dati TraceMessage

Quantità di dati del messaggio tracciati in byte per le operazioni MQGET, MQPUT, MQPUT1e Callback. Il valore predefinito è 0.

Messaggio StopOnGetTrace

Può essere impostato su ON o OFF. Il valore predefinito è ON.

SubscriptionDelivery

Può essere impostato su BATCH o IMMEDIATE. Determina se i parametri ActivityInterval e ActivityCount devono essere utilizzati quando sono presenti una o più sottoscrizioni di traccia dell'attività. L'impostazione di questo parametro su IMMEDIATE determina la sovrascrittura dei valori di ActivityInterval e ActivityCount con i valori effettivi di 1 quando i dati di traccia hanno una sottoscrizione corrispondente. Ogni record di traccia dell'attività non viene raggruppato in batch con altri record della stessa connessione e viene invece consegnato alla sottoscrizione immediatamente senza alcun ritardo. L'impostazione IMMEDIATE aumenta il sovraccarico delle prestazioni della raccolta dei dati di traccia dell'attività. L'impostazione predefinita è BATCHED.

Riferimento messaggio di traccia attività dell'applicazione

Utilizzare questa pagina per una panoramica del formato dei messaggi di traccia dell'attività dell'applicazione e delle informazioni restituite in questi messaggi

I messaggi di traccia dell'attività dell'applicazione sono messaggi IBM MQ standard che contengono un descrittore di messaggi e dati di messaggi. I dati del messaggio contengono informazioni sulle operazioni MQI eseguite dalle applicazioni IBM MQ o sulle attività che si verificano in un sistema IBM MQ.

Descrittore messaggio

- Una struttura MQMD

Dati messaggio

- Un'intestazione PCF (MQCFH)
- I dati del messaggio di traccia dell'attività dell'applicazione che vengono sempre restituiti
- Dati del messaggio di traccia dell'attività dell'applicazione specifici dell'operazione

MQMD (message descriptor) del messaggio di traccia dell'attività dell'applicazione

Utilizzare questa pagina per comprendere le differenze tra il descrittore del messaggio dei messaggi di traccia dell'attività dell'applicazione e il descrittore del messaggio dei messaggi di eventi

I parametri e i valori nel descrittore del messaggio del messaggio di traccia dell'attività dell'applicazione sono gli stessi del descrittore del messaggio dei messaggi di eventi, con la seguente eccezione:

Format

Descrizione: Nome formato dei dati del messaggio.
Valore: **MMQFMT_ADMIN**
Messaggio di amministrazione.

CorrelId

Descrizione: Identificativo di correlazione.
Valore: Inizializzato con ConnectionId dell'applicazione

MQCFH (intestazione PCF)

Utilizzare questa pagina per visualizzare i valori PCF contenuti dalla struttura di MQCFH per un messaggio di traccia dell'attività

Per un messaggio di traccia dell'attività, la struttura MQCFH contiene i seguenti valori:

Type

Descrizione: Tipo di struttura che identifica il contenuto del messaggio.
Tipo di dati: MQLONG.
Valore: ATTIVITÀ MQCFT_APP_

StrucLength

Descrizione: Lunghezza in byte della struttura MQCFH.
Tipo di dati: MQLONG.
Valore: LUNGHEZZA_STRUTTURA_MQCFH_STRUCT

Version

Descrizione: Numero di versione della struttura.
Tipo di dati: MQLONG.
Valori: MQCFH_VERSION_3

Command

Descrizione: Identificativo comando. Questo campo identifica la categoria del messaggio.
Tipo di dati: MQLONG.

Valori: MQCMD_ATTIVITA_TRACE

MsgSeqNumber

Descrizione: Numero di sequenza del messaggio. Questo campo è il numero di sequenza del messaggio all'interno di un gruppo di messaggi correlati.

Tipo di dati: MQLONG.

Valori: 1

Control

Descrizione: Opzioni di controllo.

Tipo di dati: MQLONG.

Valori: MQCFC_LAST.

CompCode

Descrizione: Codice di completamento.

Tipo di dati: MQLONG.

Valori: MQCC_OK.

Reason

Descrizione: Codice di errore che qualifica il codice di completamento.

Tipo di dati: MQLONG.

Valori: MQRC_NONE.

ParameterCount

Descrizione: Conteggio delle strutture di parametri. Questo campo è il numero di strutture di parametri che seguono la struttura MQCFH. Una struttura di gruppi (MQCFGR) e le relative strutture di parametri incluse vengono conteggiate come una sola struttura.

Tipo di dati: MQLONG.

Valori: 1 o superiore

Dati del messaggio di traccia dell'attività dell'applicazione

Immediatamente dopo l'intestazione PCF è una serie di parametri che descrivono l'intervallo di tempo per la traccia dell'attività. Questi parametri indicano anche la sequenza dei messaggi nel momento in cui vengono scritti. L'ordine e il numero dei campi che seguono l'intestazione non sono garantiti, consentendo l'aggiunta di ulteriori informazioni in futuro.

Nome messaggio: Messaggio di traccia dell'attività.

Coda di sistema: SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

QueueManager

Descrizione: Il nome del gestore code.

Identificativo: MQCA_Q_MGR_NAME

Tipo di dati: MQCFST

Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_

QSGName



Descrizione: Il nome del gruppo di condivisione code di cui il gestore code è membro (soloz/OS).

Identificativo: MQCA_QSG_NAME

Tipo di dati: MQCFST

Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_

HostName

Descrizione: Il nome host della macchina su cui è in esecuzione il gestore code.

Identificativo: MQCACF_NOME_HOST

Tipo di dati: MQCFST

IntervalStartDate

Descrizione: La data di inizio del periodo di monitoraggio.

Identificativo: DATA_STAR MQCAMO

Tipo di dati: MQCFST

Lunghezza massima: MQ_DATE_LENGTH

IntervalStartTime

Descrizione: L'ora di avvio del periodo di controllo.

Identificativo: ORA_START_MQCAMO

Tipo di dati: MQCFST

Lunghezza massima: MQ_ORA_LENGTH

IntervalEndDate

Descrizione: La data di fine del periodo di controllo.

Identificativo: DATA_FINE MQCAMO

Tipo di dati: MQCFST

Lunghezza massima: MQ_DATE_LENGTH

IntervalEndTime

Descrizione: L'ora di fine del periodo di monitoraggio.

Identificativo: ORA_END_MQCAMO

Tipo di dati: MQCFST

Lunghezza massima: MQ_ORA_LENGTH

CommandLevel

Descrizione: Il livello di comando IBM MQ .
Identificativo: LIVELLO_COMMAND_MQI
Tipo di dati: MQCFIN

SeqNumber

Descrizione: Il numero di sequenza normalmente zero. Questo valore viene incrementato per ogni successivo record per connessioni di lunga durata.
Identificativo: NUMERO_SEQUENZA_MQIACF_
Tipo di dati: MQCFIN

ApplicationName

Descrizione: Il nome della domanda (nome programma).
Identificativo: MQCACF_APPL_NAME
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_APPL_MQ

ApplClass

Descrizione: Tipo di applicazione che ha eseguito l'attività. Valori possibili: MQAT_*
Identificativo: TIPO_APPL_MQI
Tipo di dati: MQCFIN

ApplicationPid

Descrizione: L'ID processo del sistema operativo dell'applicazione.
Identificativo: ID_PROCESSO_MQIACF
Tipo di dati: MQCFIN

UserId

Descrizione: Il contesto dell'identificativo utente dell'applicazione.
Identificativo: IDENTIFICATIVO_UTENTE_MQCACF
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_ID_UTENTE_MQ

APICallerType

Descrizione: Il tipo di applicazione. Valori possibili: MQXACT_EXTERNAL o MQXACT_INTERNAL.
Identificativo: TIPO_CALLER_API_MQIACF
Tipo di dati: MQCFIN

Environment

Descrizione: L'ambiente di runtime dell'applicazione. Valori possibili: MQXE_*

Identificativo: AMBIENTE API MQIACF_AMBIENTE
Tipo di dati: MQCFIN

ChannelName

Descrizione: Il nome canale associato alla connessione. Questo parametro viene restituito solo quando il valore del parametro Environment è MQXE_MCA o MQXE_MCA_SVRCONN.
Identificativo: MQCACH_CHANNEL_NAME
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_CANALE_MQ

ConnectionName

Descrizione: Il nome della connessione di rete associato alla connessione. Questo parametro viene restituito solo quando il valore del parametro Environment è MQXE_MCA o MQXE_MCA_SVRCONN.
Identificativo: MQCACH_CONNECTION_NAME
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_CONN_MQ

ChannelType

Descrizione: Il tipo di canale associato alla connessione. Questo parametro viene restituito solo quando il valore del parametro Environment è MQXE_MCA o MQXE_MCA_SVRCONN. Valori possibili: MQCHT_*
Identificativo: TIPO_MQIACH_CHANNEL_
Tipo di dati: MQCFIN

RemoteProduct

Descrizione: L'identificativo del prodotto remoto associato alla connessione. Questo parametro viene restituito solo quando il valore del parametro Environment è MQXE_MCA o MQXE_MCA_SVRCONN.
Identificativo: MQCACH_REMOTE_PRODUCT
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_PRODOTTO_REMOTO_MQ_

RemoteVersion

Descrizione: La versione del prodotto remota associata al collegamento. Questo parametro viene restituito solo quando il valore del parametro Environment è MQXE_MCA o MQXE_MCA_SVRCONN.
Identificativo: MQCACH_REMOTE_VERSIONE
Tipo di dati: MQCFST
Lunghezza massima: LUNGHEZZA_VERSIONE_REMOTA_MQ_

FunctionName

Descrizione:	Il nome dell'ultima funzione di livello superiore avviata dal sottoprocesso iniziale.
Identificativo:	MQCACF_APPL_FUNCTION
Tipo di dati:	MQCFST

FunctionType

Descrizione:	Il tipo dell'ultima funzione di livello superiore avviata dal sottoprocesso iniziale. Valori possibili: MQFUN_*
Identificativo:	TIPO_FUNZIONE_APPL_MQIACF_
Tipo di dati:	MQCFIN

Detail

Descrizione:	Il livello di dettaglio registrato per la connessione. Valori possibili: 1=LOW 2=MEDIUM 3=HIGH
Identificativo:	DETTAGLI_MQIACF_TRACE_DETAIL
Tipo di dati:	MQCFIN

TraceDataLength

Descrizione:	La lunghezza dei dati del messaggio (in byte) tracciata per questa connessione.
Identificativo:	LUNGHEZZA_DATI_TRACCIA_MQIACF
Tipo di dati:	MQCFIN

PointerSize

Descrizione:	La lunghezza (in byte) dei puntatori sulla piattaforma che l'applicazione sta eseguendo (per assistere nell'interpretazione delle strutture binarie).
Identificativo:	DIMENSIONE_POINTER_MQIACF
Tipo di dati:	MQCFIN

Platform

Descrizione:	La piattaforma su cui il gestore code è in esecuzione. Valori possibili: MQPL_*
Identificativo:	PIATFORM_MQIA
Tipo di dati:	MQCFIN

Parametri variabili per le operazioni MQI dell'attività dell'applicazione

La struttura MQCFGR dei dati dell'attività dell'applicazione è seguita dalla serie di parametri PCF che corrispondono all'operazione che si sta eseguendo. I parametri per ciascuna operazione sono definiti nella seguente sezione.

Il livello di traccia indica il livello di granularità della traccia richiesto per i parametri da includere nella traccia. I valori possibili del livello di traccia sono:

1. Bassa

Il parametro viene incluso quando la traccia di attività "bassa", "media" o "alta" è configurata per un'applicazione. Questa impostazione indica che un parametro è sempre incluso nel gruppo AppActivityData per l'operazione. Questa serie di parametri è sufficiente per tenere traccia delle chiamate MQI effettuate da un'applicazione e per verificare se hanno esito positivo.

2. Medio

Il parametro è incluso nel gruppo AppActivityData per l'operazione solo quando la traccia "media" o "alta" attività è configurata per un'applicazione. Questa serie di parametri aggiunge informazioni sulle risorse, ad esempio, i nomi di coda e argomento utilizzati dall'applicazione.

3. Elevato

Il parametro è incluso solo nel gruppo AppActivityData per l'operazione quando la traccia "alta" attività è configurata per un'applicazione. Questa serie di parametri include i dump di memoria delle strutture trasmesse alle funzioni MQI e XA. Per questo motivo, contiene ulteriori informazioni sui parametri utilizzati nelle chiamate MQI e XA. I dump di memoria della struttura sono copie poco profonde delle strutture. Per evitare tentativi errati di annullare il riferimento dei puntatori, i valori del puntatore nelle strutture sono impostati su NULL.

Nota: La versione della struttura di cui viene eseguito il dump non è necessariamente identica alla versione utilizzata da una applicazione. La struttura può essere modificata da un'uscita incrociata API, dal codice di traccia dell'attività o dal gestore code. Un gestore code può modificare una struttura in una versione successiva, ma il gestore code non la modifica mai in una versione precedente della struttura. Per farlo, rischierebbe di perdere dati.

MQBACK

L'applicazione ha avviato la funzione MQBACK MQI

CompCode

Descrizione:	Il codice di completamento che indica il risultato dell'operazione
Parametro PCF:	CODICE_COMP_MQIACF
Livello di traccia:	1
Tipo	MQCFIN

Reason

Descrizione:	Il risultato del codice motivo dell'operazione
Parametro PCF:	CODICE_REASON_MQIACF
Livello di traccia:	1
Tipo	MQCFIN

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF:	MQIAMO64_QMGR_OP_DURATION
Livello di traccia:	2
Tipo	MQCFIN64

MQBEGIN

L'applicazione ha avviato la funzione MQBEGIN MQI

CompCode

Descrizione:	Il codice di completamento che indica il risultato dell'operazione
--------------	--

Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo MQCFIN

MQBO

Descrizione: La struttura delle opzioni MQBEGIN. Questo parametro non viene incluso se viene utilizzato un puntatore NULL nella chiamata MQBEGIN.
Parametro PCF: MQBACF_MQBO_STRUCT
Livello di traccia: 3
Tipo MQCFB
Lunghezza: La lunghezza in byte della struttura MQBO.

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.
La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .
Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.
Parametro PCF: MQIAMO64_QMGR_OP_DURATION
Livello di traccia: 2
Tipo MQCFIN64

MQCALLBACK

L'applicazione ha avviato la funzione MQCALLBACK

ObjectHandle

Descrizione: L'handle dell'oggetto
Parametro PCF: HOBJ MQIACF
Livello di traccia: 1
Tipo MQCFIN

CallType

Descrizione: Perché è stata richiamata la funzione. Uno dei valori MQCBCT_*
Parametro PCF: TIPO_CALL_MQIACF
Livello di traccia: 1
Tipo MQCFIN

MsgBuffer

Descrizione:	Dati del messaggio.
Parametro PCF:	DATI_MESSAGE_MQBACF
Livello di traccia:	1
Tipo	MQCFB
Lunghezza:	La lunghezza è regolata dal parametro TRACEDATA () impostato nella configurazione APPTRACE. Se TRACEDATA=NONE, questo parametro viene omesso.

MsgLength

Descrizione:	Lunghezza del messaggio. (Preso dal campo DataLength nella struttura MQCBC).
Parametro PCF:	MQIACF_MSG_LENGTH
Livello di traccia:	1
Tipo	MQCFIN

HighResTime

Descrizione:	Ora dell'operazione in microsecondi dalla mezzanotte del 1 gennaio 1st 1970 (UTC) Nota: L'accuratezza di questo timer varia in base al supporto della piattaforma per un timer ad alta risoluzione
Parametro PCF:	MQIAMO64_HIGHRES_TIME
Livello di traccia:	2
Tipo	MQCFIN64

ReportOptions

Descrizione:	Opzioni per messaggi di report
Parametro PCF:	REPORT MQIACF
Livello di traccia:	2
Tipo	MQCFIN

MsgType

Descrizione:	Tipo di messaggio
Parametro PCF:	TIPO_MSG_MQIACF
Livello di traccia:	2
Tipo	MQCFIN

Expiry

Descrizione:	Durata messaggio
Parametro PCF:	MQIACF_SCADENZA
Livello di traccia:	2
Tipo	MQCFIN

Format

Descrizione:	Nome formato dei dati di messaggio
Parametro PCF:	MQCACH_FORMAT_NAME
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	MQ_FORMAT_LENGTH

Priority

Descrizione:	Priorità messaggio
Parametro PCF:	PRIORITÀ_MQIACF
Livello di traccia:	2
Tipo	MQCFIN

Persistence

Descrizione:	Persistenza messaggio
Parametro PCF:	MQIACF_PERSISTENZA
Livello di traccia:	2
Tipo	MQCFIN

MsgId

Descrizione:	ID messaggio
Parametro PCF:	ID_MSG_MQBACF
Livello di traccia:	2
Tipo	MQCFB
Lunghezza:	ID_MQ_MSG_LENGTH

CorrelId

Descrizione:	Identificativo di correlazione
Parametro PCF:	ID_CORREL_MQBACF
Livello di traccia:	2
Tipo	MQCFB
Lunghezza:	LUNGHEZZA_ID_CORRELATA

ObjectName

Descrizione:	Il nome dell'oggetto aperto.
Parametro PCF:	MQCACF_XX_ENCODE_CASE_ONE nome_oggetto
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	LUNGHEZZA_NOME_Q_MQ

ResolvedQName

Descrizione: Il nome locale della coda da cui è stato richiamato il messaggio.
Parametro PCF: MQCACF_RESOLVED_Q_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

ReplyToQueue

Descrizione: LUNGHEZZA_NOME_Q_MQ
Parametro PCF: MQCACF_REPLY_TO_Q
Livello di traccia: 2
Tipo: MQCFST

ReplyToQMgr

Descrizione: LUNGHEZZA_NOME_MQ_Q_MGR_
Parametro PCF: MQCACF_REPLY_TO_Q_MGR
Livello di traccia: 2
Tipo: MQCFST

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio
Parametro PCF: ID_MQIA_CODED_CHAR_SET_
Livello di traccia: 2
Tipo: MQCFIN

Encoding

Descrizione: Codifica numerica dei dati del messaggio.
Parametro PCF: MQIACF_ENCODING
Livello di traccia: 2
Tipo: MQCFIN

PutDate

Descrizione: LUNGHEZZA_DATA_PUT_MQ
Parametro PCF: DATA_PUT_MQCACF
Livello di traccia: 2
Tipo: MQCFST

PutTime

Descrizione: LUNGHEZZA_TEMPO_PUT_MQ
Parametro PCF: ORA_PUT_MQCACF
Livello di traccia: 2

Tipo MQCFST

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.

Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME

Livello di traccia: 2

Tipo MQCFST

Lunghezza: LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione: Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.

Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING

Livello di traccia: 2

Tipo MQCFST

Lunghezza: La lunghezza varia.

ResolvedType

Descrizione: Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.

Parametro PCF: TIPO_RESOLV_MQIACF

Livello di traccia: 2

Tipo MQCFIN

PolicyName

Descrizione: Il nome della politica applicato a questo messaggio.

Nota: Solo messaggi protetti AMS

Parametro PCF: NOME_POLITICA_MQCA

Livello di traccia: 2

Tipo MQCFST

Lunghezza: LUNGHEZZA_NOME_OGGETTO_MQ

XmitqMsgId

Descrizione: L'ID del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: ID_MSG_XQH_MQBACF

Livello di traccia: 2

Tipo MQCFB

Lunghezza: ID_MQ_MSG_LENGTH

XmitqCorrelId

Descrizione:	L'ID di correlazione del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQBACF_XQH_CORREL_ID
Livello di traccia:	2
Tipo	MQCFB
Lunghezza:	LUNGHEZZA_ID_CORRELATA

XmitqPutTime

Descrizione:	L'ora di immissione del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_PUT_TIME
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	LUNGHEZZA_TEMPO_PUT_MQ

XmitqPutDate

Descrizione:	La data di inserimento del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_PUT_DATE
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	LUNGHEZZA_DATA_PUT_MQ

XmitqRemoteQName

Descrizione:	La destinazione della coda remota del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_REMOTE_Q_Name
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	LUNGHEZZA_NOME_Q_MQ

XmitqRemoteQMgr

Descrizione:	L'ID del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_REMOTE_Q_MGR
Livello di traccia:	2
Tipo	MQCFST

Lunghezza: ID_MQ_MSG_LENGTH

MsgDescStructure

Descrizione: La struttura MQMD. Questo parametro viene omissso se viene utilizzato un MQGMO versione 4 per richiedere la restituzione di un handle del messaggio invece di un MQMD

Parametro PCF: MQBACF_MQMD_STRUCT

Livello di traccia: 3

Tipo MQCFB

Lunghezza: La lunghezza in byte della struttura MQMD (la dimensione effettiva dipende dalla versione della struttura)

GetMsgOptsStructure

Descrizione: La struttura MQGMO.

Parametro PCF: MQBACF_MQGMO_STRUCT

Livello di traccia: 3

Tipo MQCFB

Lunghezza: La lunghezza in byte della struttura MQGMO (la dimensione effettiva dipende dalla versione della struttura)

MQCBCContextStructure

Descrizione: La struttura MQCBC.

Parametro PCF: MQBACF_MQCBC_STRUCT

Livello di traccia: 3

Tipo MQCFB

Lunghezza: La lunghezza in byte della struttura MQCBC (la dimensione effettiva dipende dalla versione della struttura)

QMgrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo MQCFIN64

MQCB

L'applicazione ha avviato la funzione MQI di gestione callback

CallbackOperation

Descrizione: L'operazione di gestione della funzione di richiamata. Impostare su uno dei valori MQOP_*

Parametro PCF: OPERAZIONE MQIACF_MQCB
Livello di traccia: 1
Tipo MQCFIN

CallbackType

Descrizione: Il tipo della funzione callback (campoCallbackType dalla struttura MQCBD). Impostare su uno dei valori MQCBT_*
Parametro PCF: TIPO_MQCB_MQIACF
Livello di traccia: 1
Tipo MQCFIN

CallbackOptions

Descrizione: Le opzioni di callback. Impostare su uno dei valori MQCBDO_*
Parametro PCF: MQIACF_MQCB_OPZIONI
Livello di traccia: 1
Tipo MQCFIN

CallbackFunction

Descrizione: Il puntatore alla funzione di callback se avviato come una chiamata di funzione.
Parametro PCF: MQBACF_MQCB_FUNZIONE
Livello di traccia: 1
Tipo MQCFB
Lunghezza: Dimensione di MQPTR

CallbackName

Descrizione: Il nome della funzione di callback se avviata come un programma collegato dinamicamente.
Parametro PCF: MQCACF_MQCB_NAME
Livello di traccia: 1
Tipo MQCFST
Lunghezza: Dimensione di MQCHAR128

ObjectHandle

Descrizione: L'handle dell'oggetto
Parametro PCF: HOBJ MQIACF
Livello di traccia: 1
Tipo MQCFIN

MaxMsgLength

Descrizione: la lunghezza massima del messaggio. Impostare su un numero intero o sul valore speciale MQCBD_FULL_MSG_LENGTH
Parametro PCF: MQIACH_MAX_MSG_LENGTH

Livello di traccia: 2
Tipo MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo MQCFIN

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia: 2
Tipo MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione: Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia: 2
Tipo MQCFST
Lunghezza: La lunghezza varia.

ResolvedType

Descrizione: Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF: TIPO_RESOLV_MQIACF
Livello di traccia: 2
Tipo MQCFIN

Callback DescriptorStructure

Descrizione: La struttura MQCBD. Questo parametro viene omesso se alla chiamata MQCB viene passato un valore MQCBC NULL.
Parametro PCF: MQBACF_MQCBD_STRUCT
Livello di traccia: 3

Tipo MQCFB
Lunghezza: La lunghezza in byte della struttura MQCBC

MsgDescStructure

Descrizione: La struttura MQMD. Il parametro MsgDescStructure viene omissso se un valore MQMD NULL viene passato alla chiamata MQCB.
Parametro PCF: MQBACF_MQMD_STRUCT
Livello di traccia: 3
Tipo MQCFB
Lunghezza: La lunghezza in byte della struttura MQMD (la dimensione effettiva dipende dalla versione della struttura)

GetMsgOptsStructure

Descrizione: La struttura MQGMO. Questo parametro viene omissso se un valore MQGMO NULL viene passato alla chiamata MQCB.
Parametro PCF: MQBACF_MQGMO_STRUCT
Livello di traccia: 3
Tipo MQCFB
Lunghezza: La lunghezza in byte della struttura MQGMO (la dimensione effettiva dipende dalla versione della struttura)

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.
La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .
Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.
Parametro PCF: MQIAMO64_QMGR_OP_DURATION
Livello di traccia: 2
Tipo MQCFIN64

MQCLOSE

L'applicazione ha avviato la funzione MQCLOSE

ObjectHandle

Descrizione: L'handle dell'oggetto
Parametro PCF: HOBJ MQIACF
Livello di traccia: 1
Tipo MQCFIN

CloseOptions

Descrizione: Opzioni di chiusura
Parametro PCF: OPZIONI_CHIUDI_MQIACF

Livello di traccia: 1
Tipo MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo MQCFIN

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia: 2
Tipo MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione: Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia: 2
Tipo MQCFST
Lunghezza: La lunghezza varia.

ResolvedType

Descrizione: Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF: TIPO_RESOLV_MQIACF
Livello di traccia: 2
Tipo MQCFIN

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo MQCFIN64

MQCMIT

L'applicazione ha avviato la funzione MQCMIT MQI

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione

Parametro PCF: CODICE_COMP_MQIACF

Livello di traccia: 1

Tipo MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione

Parametro PCF: CODICE_REASON_MQIACF

Livello di traccia: 1

Tipo MQCFIN

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo MQCFIN64

MQCONN e MQCONNX

L'applicazione ha avviato la funzione MQCONN o MQCONNX MQI

ConnectionId

Descrizione: L'ID connessione, se disponibile, oppure MQCONNID_NONE, se non disponibile

Parametro PCF: ID_CONNESSIONE_MQBACF

Livello di traccia: 1

Tipo: MQCFB

Lunghezza massima: LUNGHEZZA_ID_CONNESSIONE_MQ

QueueManagerName

Descrizione: Il nome (non risolto) del gestore code utilizzato nella chiamata MQCONN (X)
Parametro PCF: MQCA_Q_MGR_NAME
Livello di traccia: 1
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

ConnectOptions

Descrizione: Opzioni di connessione derivate dai valori MQCNO_*
Nota: Solo MQCONNX
Parametro PCF: OPZIONI_MQIACF_CONNECT_
Livello di traccia: 2
Tipo: MQCFIN

ConnectionOptionsStructure

Descrizione: La struttura MQCNO.
Nota: solo MQCONNX)
Parametro PCF: MQBACF_MQCNO_STRUCT
Livello di traccia: 3
Tipo: MQCFB
Lunghezza massima: La lunghezza in byte della struttura MQCNO (la dimensione effettiva dipende dalla versione della struttura)

ChannelDefinitionStructure

Descrizione: La struttura MQCD.
Nota: Solo connessioni client

Parametro PCF:	MQBACF_MQCD_STRUCT
Livello di traccia:	3
Tipo:	MQCFB
Lunghezza massima:	La lunghezza in byte della struttura MQCD (la dimensione effettiva dipende dalla versione della struttura)

QMgrOpDuration

Descrizione:	Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code. La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ . Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.
Parametro PCF:	MQIAMO64_QMGR_OP_DURATION
Livello di traccia:	2
Tipo	MQCFIN64

MQCTL

L'applicazione ha avviato la funzione MQI di MQCTL

CompCode

Descrizione:	Il codice di completamento che indica il risultato dell'operazione
Parametro PCF:	CODICE_COMP_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

Reason

Descrizione:	Il risultato del codice motivo dell'operazione
Parametro PCF:	CODICE_REASON_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

CtlOperation

Descrizione:	Uno dei valori MQOP_*
Parametro PCF:	OPERAZIONE MQIACF_CTL
Livello di traccia:	1
Tipo:	MQCFIN

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo: MQCFIN64

MQDISC

L'applicazione ha avviato la funzione MQDISC MQI

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione

Parametro PCF: CODICE_COMP_MQIACF

Livello di traccia: 1

Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione

Parametro PCF: CODICE_REASON_MQIACF

Livello di traccia: 1

Tipo: MQCFIN

MQGET

L'applicazione ha avviato la funzione MQGET MQI

ObjectHandle

Descrizione: L'handle dell'oggetto

Parametro PCF: HOBJ MQIACF

Livello di traccia: 1

Tipo: MQCFIN

GetOptions

Descrizione: Le opzioni get da MQGMO.Options

Parametro PCF: OPZIONI MQIACF_GET

Livello di traccia: 1

Tipo: MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione

Parametro PCF: CODICE_COMP_MQIACF

Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

MsgBuffer

Descrizione: Dati del messaggio. Se TRACEDATA=NONE, questo parametro viene omissso
Parametro PCF: DATI_MESSAGE_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza massima: La lunghezza è regolata dal parametro TRACEDATA () impostato nella configurazione APPTRACE. (Incluso nel messaggio di traccia come MQIACF_TRACE_DATA_LENGTH).

MsgLength

Descrizione: Lunghezza del messaggio.
Parametro PCF: MQIACF_MSG_LENGTH
Livello di traccia: 1
Tipo: MQCFIN

HighResTime

Descrizione: Ora di funzionamento in microsecondi dalla mezzanotte del 1 gennaio 1970 (UTC)
Nota: L'accuratezza di questo timer varia in base al supporto della piattaforma per un timer ad alta risoluzione
Parametro PCF: MQIAMO64_HIGHRES_TIME
Livello di traccia: 2
Tipo: MQCFIN64

BufferLength

Descrizione: Lunghezza del buffer fornito dall'applicazione
Parametro PCF: LUNGHEZZA_BUFFER_MQIACF_
Livello di traccia: 2
Tipo: MQCFIN

ObjectName

Descrizione: Il nome dell'oggetto aperto
Parametro PCF: MQCACF_XX_ENCODE_CASE_ONE nome_oggetto

Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

ResolvedQName

Descrizione: Il nome locale della coda da cui è stato richiamato il messaggio.
Parametro PCF: MQCACF_RESOLVED_Q_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_Q_MQ

ReportOptions

Descrizione: Opzioni di report dei messaggi
Parametro PCF: REPORT MQIACF
Livello di traccia: 2
Tipo: MQCFIN

MsgType

Descrizione: Tipo di messaggio
Parametro PCF: TIPO_MSG_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

Expiry

Descrizione: Durata messaggio
Parametro PCF: MQIACF_SCADENZA
Livello di traccia: 2
Tipo: MQCFIN

Format

Descrizione: Nome formato dei dati di messaggio
Parametro PCF: MQCACH_FORMAT_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: MQ_FORMAT_LENGTH

Priority

Descrizione: Priorità messaggio
Parametro PCF: PRIORITÀ_MQIACF
Livello di traccia: 2

Tipo: MQCFIN

Persistence

Descrizione: Persistenza messaggio
Parametro PCF: MQIACF_PERSISTENZA
Livello di traccia: 2
Tipo: MQCFIN

MsgId

Descrizione: ID messaggio
Parametro PCF: ID_MSG_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza massima: ID_MQ_MSG_LENGTH

CorrelId

Descrizione: Identificativo di correlazione
Parametro PCF: ID_CORREL_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza massima: LUNGHEZZA_ID_CORRELATA

ReplyToQueue

Descrizione:
Parametro PCF: MQCACF_REPLY_TO_Q
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_Q_MQ

ReplyToQMgr

Descrizione:
Parametro PCF: MQCACF_REPLY_TO_Q_MGR
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio
Parametro PCF: ID_MQIA_CODED_CHAR_SET_

Livello di traccia: 2
Tipo: MQCFIN

Encoding

Descrizione: Codifica numerica dei dati del messaggio.
Parametro PCF: MQIACF_ENCODING
Livello di traccia: 2
Tipo: MQCFIN

PutDate

Descrizione:
Parametro PCF: DATA_PUT_MQCACF
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_DATA_PUT_MQ

PutTime

Descrizione:
Parametro PCF: ORA_PUT_MQCACF
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_TEMPO_PUT_MQ

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione: Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: La lunghezza varia.

ResolvedType

Descrizione:	Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF:	TIPO_RESOLV_MQIACF
Livello di traccia:	2
Tipo	MQCFIN

PolicyName

Descrizione:	Il nome della politica applicato a questo messaggio. Nota: Solo messaggi protetti AMS
Parametro PCF:	NOME_POLITICA_MQCA
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_NOME_OGGETTO_MQ

XmitqMsgId

Descrizione:	L'ID del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	ID_MSG_XQH_MQBACF
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza:	ID_MQ_MSG_LENGTH

XmitqCorrelId

Descrizione:	L'ID di correlazione del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQBACF_XQH_CORREL_ID
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza:	LUNGHEZZA_ID_CORRELATA

XmitqPutTime

Descrizione:	L'ora di immissione del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_PUT_TIME
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_TEMPO_PUT_MQ

XmitqPutDate

Descrizione: La data di inserimento del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_PUT_DATE

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_DATA_PUT_MQ

XmitqRemoteQName

Descrizione: La destinazione della coda remota del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_REMOTE_Q_NAME

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_Q_MQ

XmitqRemoteQMGr

Descrizione: La destinazione del gestore code remoto del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_REMOTE_Q_MGR

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_Q_MQ

MsgDescStructure

Descrizione: La struttura MQMD.

Parametro PCF: MQBACF_MQMD_STRUCT

Livello di traccia: 3

Tipo: MQCFB

Lunghezza massima: La lunghezza in byte della struttura MQMD (la dimensione effettiva dipende dalla versione della struttura)

GetMsgOptsStructure

Descrizione: La struttura MQGMO.

Parametro PCF: MQBACF_MQGMO_STRUCT

Livello di traccia: 3

Tipo: MQCFB

Lunghezza massima: La lunghezza in byte della struttura MQGMO (la dimensione effettiva dipende dalla versione della struttura)

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo: MQCFIN64

MQINQ

L'applicazione ha avviato la funzione MQINQ MQI

ObjectHandle

Descrizione: L'handle dell'oggetto

Parametro PCF: HOBJ MQIACF

Livello di traccia: 1

Tipo: MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione

Parametro PCF: CODICE_COMP_MQIACF

Livello di traccia: 1

Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione

Parametro PCF: CODICE_REASON_MQIACF

Livello di traccia: 1

Tipo: MQCFIN

SelectorCount

Descrizione: Il numero di selettori forniti nell'array Selettori.

Parametro PCF: MQIACF_SELECTOR_COUNT

Livello di traccia: 2

Tipo: MQCFIN

Selectors

Descrizione: L'elenco di attributi (numero intero o carattere) i cui valori devono essere restituiti da MQINQ.

Parametro PCF: SELECTORS MQIACF
Livello di traccia: 2
Tipo: MQCFIL

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF: MQCACF_RESOLVED_Q_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_NOME_Q_MQ

ResObjectString

Descrizione: Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: La lunghezza varia

ResolvedType

Descrizione: Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF: TIPO_RESOLV_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

IntAttrCount

Descrizione: Il numero di attributi interi restituiti dall'operazione di interrogazione
Parametro PCF: Conteggio_INTATTR_MQIACF_
Livello di traccia: 3
Tipo: MQCFIN

IntAttrS

Descrizione: I valori dell'attributo integer restituiti dall'operazione inquire. Questo parametro è presente solo se IntAttrCount è > 0 quando MQINQ viene restituito.
Parametro PCF: MQIACF_INT_ATTRS
Livello di traccia: 3
Tipo: MQCFIL

CharAttrrs

Descrizione:	Gli attributi carattere restituiti dall'operazione di interrogazione. I valori sono concatenati insieme. Questo parametro è incluso solo se la lunghezza CharAttrrè > 0 quando MQINQ viene restituito.
Parametro PCF:	MQCACF_CHAR_ATTRS
Livello di traccia:	3
Tipo:	MQCFST

QMgrOpDuration

Descrizione:	Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code. La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ . Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.
Parametro PCF:	MQIAMO64_QMGR_OP_DURATION
Livello di traccia:	2
Tipo	MQCFIN64

MQOPEN

L'applicazione ha avviato la funzione MQOPEN MQI

ObjectType

Descrizione:	Il tipo di oggetto passato in MQOT.ObjectType
Parametro PCF:	TIPO_OGGETTO_MQIAC
Livello di traccia:	1
Tipo:	MQCFIN

ObjectName

Descrizione:	Il nome dell'oggetto passato alla chiamata MQI prima che venga tentata qualsiasi risoluzione del nome della coda.
Parametro PCF:	MQCACF_XX_ENCODE_CASE_ONE nome_oggetto
Livello di traccia:	1
Tipo:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ

ObjectQMgrName

Descrizione:	Il nome del gestore code oggetti passato alla chiamata MQI prima che venga tentata qualsiasi risoluzione del nome coda.
Parametro PCF:	MQCACF_OBJECT_Q_MGR_NAME
Livello di traccia:	1
Tipo:	MQCFST

Lunghezza massima: LUNGHEZZA_NOME_MQ_Q_MGR_

ObjectHandle

Descrizione: L'handle dell'oggetto
Parametro PCF: HOBJ MQIACF
Livello di traccia: 1
Tipo: MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

OpenOptions

Descrizione: Opzioni utilizzate per aprire l'oggetto
Parametro PCF: OPEN_MQIACF_OPZIONI
Livello di traccia: 1
Tipo: MQCFIN

AlternateUserId

Descrizione: Incluso solo se è specificato MQOO_ALTERNATE_USER_AUTHORITY
Parametro PCF: MQCACF_ALTERNATE_USERID
Livello di traccia: 2
Tipo: MQCFST
Lunghezza massima: LUNGHEZZA_ID_UTENTE_MQ

RecsPresent

Descrizione: Il numero di record nome oggetto presenti. Incluso solo se MQOD Versione > = MQOD_VERSION_2
Parametro PCF: MQIACF_RECS_PRESENTE
Livello di traccia: 1
Tipo: MQCFIN

KnownDestCount

Descrizione:	Numero di code locali aperte correttamente Solo incluse se MQOD Versione > = MQOD_VERSION_2
Parametro PCF:	CONTEGGIO_DEST_CONOSCIUTO_MQIACF_
Livello di traccia:	1
Tipo:	MQCFIN

UnknownDestCount

Descrizione:	Numero di code remote aperte correttamente Solo incluse se MQOD Versione > = MQOD_VERSION_2
Parametro PCF:	MQIACF_UNKNOWN_DEST_COUNT
Livello di traccia:	1
Tipo:	MQCFIN

InvalidDestCount

Descrizione:	Numero di code che non è stato possibile aprire solo se MQOD Versione > = MQOD_VERSION_2
Parametro PCF:	MQIACF_INVALID_DEST_COUNT
Livello di traccia:	1
Tipo:	MQCFIN

DynamicQName

Descrizione:	Il nome della coda dinamica passato come input alla chiamata MQOPEN.
Parametro PCF:	MQCACF_DYNAMIC_Q_NAME
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ

ResolvedLocalQName ^{1 2}

Descrizione:	Contiene il nome della coda locale dopo che è stata eseguita la risoluzione del nome. (ad esempio, per le code remote questo sarà il nome della coda di trasmissione)
Parametro PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia:	2
Tipo:	MQCFST
Intervallo:	Se MQOD.Version è precedente a MQOD_VERSION_3 e contiene il valore di MQOD.ObjectName dopo il completamento della chiamata MQOPEN. Se MQOD.Version è uguale o superiore a MQOD_VERSION_3 e contiene il valore in MQOD. Campo ResolvedQName .
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ

ResolvedLocalQMgrName ¹ ²

Descrizione:	Il nome del gestore code locale dopo che è stata eseguita la risoluzione del nome.
Parametro PCF:	MQCACF_RESOLVED_LOCAL_Q_MGR
Livello di traccia:	2
Tipo:	MQCFST
Intervallo:	Solo se MQOD.Version > = MQOD_VERSION_3
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_

ResolvedQName ¹ ²

Descrizione:	Il nome della coda dopo la risoluzione del nome è stato eseguito.
Parametro PCF:	MQCACF_RESOLVED_Q_NAME
Livello di traccia:	2
Tipo:	MQCFST
Intervallo:	Se MQOD.Version è precedente a MQOD_VERSION_3 e contiene il valore di MQOD.ObjectName dopo il completamento della chiamata MQOPEN. Se MQOD.Version è uguale o superiore a MQOD_VERSION_3 e contiene il valore in MQOD. Campo ResolvedQName .
Lunghezza massima:	LUNGHEZZA_NOME_Q_MQ

ResolvedQMgrName ¹ ²

Descrizione:	Contiene il nome del gestore code dopo che è stata eseguita la risoluzione del nome. Se MQOD.Version è precedente a MQOD_VERSION_3 e contiene il valore di MQOD. Campo ObjectQMgrNome dopo il completamento della chiamata MQOPEN. Se MQOD.Version è uguale o superiore a MQOD_VERSION_3 e contiene il valore in MQOD. Campo ResolvedQMgrNome.
Parametro PCF:	MQCACF_RESOLVED_Q_MGR
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza massima:	LUNGHEZZA_NOME_MQ_Q_MGR_

AlternateSecurityId

Descrizione:	Identificativo di sicurezza alternativo. Presente solo se MQOD.Version è uguale o superiore a MQOD_VERSION_3, viene specificato MQOO_ALTERNATE_USER_AUTHORITY e MQOD.AlternateSecurityId non è uguale a MQSID_NONE.
Parametro PCF:	ID_SICUREZZA_ALTERNATO_MQBACF
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza massima:	LUNGHEZZA_ID_SICUREZZA_MQ

ObjectString

Descrizione:	Nome oggetto lungo. Incluso solo se MQOD.Version è uguale o superiore a MQOD_VERSION_4 e al campo VSLength di MQOD.ObjectString è MQVS_NULL_TERMINATED o è maggiore di zero.
Parametro PCF:	MQCACF_STRINGA_OGGETTO
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza massima:	La lunghezza varia.

SelectionString

Descrizione:	Stringa di selezione. Incluso solo se MQOD.Version è uguale o superiore a MQOD_VERSION_4 e al campo VSLength di MQOD.SelectionString è MQVS_NULL_TERMINATED o maggiore di zero.
Parametro PCF:	MQCACF_SELECTION_STRING
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza massima:	La lunghezza varia.

ResObjectString

Descrizione:	Il nome oggetto lungo dopo che il gestore code ha risolto il nome fornito nel campo ObjectName . Incluso solo per gli argomenti e gli alias della coda che fanno riferimento a un oggetto argomento se MQOD.Version è uguale o superiore a MQOD_VERSION_4 e VSLength è MQVS_NULL_TERMINATED o superiore a zero.
Parametro PCF:	MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza massima:	La lunghezza varia.

ResolvedType

Descrizione:	Il tipo di oggetto risolto (base) che si sta aprendo. Incluso solo se MQOD.Version è uguale o superiore a MQOD_VERSION_4. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF:	TIPO_RESOLV_MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

QMGrOpDuration

Descrizione:	Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code. La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ . Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.
Parametro PCF:	MQIAMO64_QMGR_OP_DURATION
Livello di traccia:	2
Tipo	MQCFIN64

Struttura intestazione gruppo PCF elenco di distribuzione attività applicazione

Se la funzione MQOPEN apre un elenco di distribuzione, allora i parametri MQOPEN includono un gruppo PCF AppActivityDistList per ciascuna delle code nell'elenco di distribuzione fino al numero di strutture numerate in RecsPresent. The AppActivityDistList PCF group combines information from the MQOR, and MQRR structures to identify the queue name, and indicate the result of the open operation on the queue. Un gruppoDistList di AppActivityinizia sempre con la struttura MQCFGR seguente:

<i>Tabella 29. Struttura MQCFGR gruppo AppActivityDistList</i>		
campo MQCFGR	Valore	Descrizione
Tipo	Gruppo_MQCF	
StrucLength	Lunghezza in byte della struttura MQCFGR	
Parametro	ELENCO DIST_APP MQGACF	Parametro gruppo elenco di distribuzione
ParameterCount	4	Il numero di strutture di parametri che seguono la struttura MQCFGR contenute in questo gruppo.

ObjectName

Descrizione:	Il nome di una coda nell'elenco di distribuzione MQ_Q_NAME_LENGTH. Incluso solo se vengono fornite le strutture MQOR.
Parametro PCF:	MQCACF_XX_ENCODE_CASE_ONE nome_oggetto
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_NOME_Q_MQ. Incluso solo se vengono fornite le strutture MQOR.

ObjectQMGrName

Descrizione:	Il nome del gestore code su cui è definita la coda denominata in ObjectName .
Parametro PCF:	MQCACF_OBJECT_Q_MGR_NAME

¹ Questo parametro viene incluso solo se l'oggetto da aprire si risolve in una coda e la coda viene aperta per MQOO_INPUT_*, MQOO_OUTPUT o MQOO_BROWSE

² Il parametro QName ResolvedLocalè incluso solo se è diverso dal parametro ResolvedQName .

Livello di traccia: 2
Tipo: MQCFST
Lunghezza: MQ_Q_MGR_NAME_LENGTH. Incluso solo se vengono fornite le strutture MQOR.

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'apertura per questo oggetto. Incluso solo se vengono fornite le strutture MQRR e il codice motivo per MQOPEN è MQRC_MULTIPLE_REASON
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

Reason

Descrizione: Il codice motivo che indica il risultato dell'apertura per questo oggetto. Incluso solo se vengono fornite le strutture MQRR e il codice motivo per MQOPEN è MQRC_MULTIPLE_REASON
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

MQPUT

L'applicazione ha avviato la funzione MQPUT MQI.

ObjectHandle

Descrizione: L'handle dell'oggetto
Parametro PCF: HOBJ MQIACF
Livello di traccia: 1
Tipo: MQCFIN

PutOptions

Descrizione: Le opzioni put da MQPMO.Options
Parametro PCF: OPZIONI MQIACF_PUT_
Livello di traccia: 1
Tipo: MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione

Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

MsgBuffer

Descrizione: Dati del messaggio.
Parametro PCF: DATI_MESSAGE_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: La lunghezza è regolata dal parametro TRACEDATA () impostato nella configurazione APPTRACE. Se TRACEDATA=NONE, questo parametro viene omissso.

MsgLength

Descrizione: Lunghezza del messaggio.
Parametro PCF: MQIACF_MSG_LENGTH
Livello di traccia: 1
Tipo: MQCFIN

RecsPresent

Descrizione: Il numero di record di messaggi di inserimento o di risposta presenti. Incluso solo se MQPMO Versione > = MQPMO_VERSION_2
Parametro PCF: MQIACF_RECS_PRESENTE
Livello di traccia: 1
Tipo: MQCFIN

KnownDestCount

Descrizione: Numero di messaggi inviati correttamente alle code locali
Parametro PCF: CONTEGGIO_DEST_CONOSCIUTO_MQIACF_
Livello di traccia: 1
Tipo: MQCFIN

UnknownDestCount

Descrizione: Numero di messaggi inviati correttamente alle code remote
Parametro PCF: MQIACF_UNKNOWN_DEST_COUNT
Livello di traccia: 1
Tipo: MQCFIN

InvalidDestCount

Descrizione: Numero di messaggi che non è stato possibile inviare
Parametro PCF: MQIACF_INVALID_DEST_COUNT
Livello di traccia: 1

Tipo: MQCFIN

HighResTime

Descrizione: Ora dell'operazione in microsecondi dalla mezzanotte del 1 gennaio 1st 1970 (UTC)

Nota: La precisione di questo timer varia in base al supporto della piattaforma per un timer ad elevata risoluzione.

Parametro PCF: MQIAMO64_HIGHRES_TIME

Livello di traccia: 2

Tipo: MQCFIN64

ObjectName

Descrizione: Il nome dell'oggetto aperto.

Parametro PCF: MQCACF_XX_ENCODE_CASE_ONE nome_oggetto

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_Q_MQ

ResolvedQName

Descrizione: Il nome della coda dopo che è stata eseguita la risoluzione del nome della coda.

Parametro PCF: MQCACF_RESOLVED_Q_NAME

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_Q_MQ

ResolvedQMgrName

Descrizione: Il nome del gestore code dopo che è stata eseguita la risoluzione del nome.

Parametro PCF: MQCACF_RESOLVED_Q_MGR

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_MQ_Q_MGR_

ResolvedLocalQName³

Descrizione: Contiene il nome della coda locale dopo che è stata eseguita la risoluzione del nome.

Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME

Livello di traccia: 2

Tipo: MQCFST

ResolvedLocalQMgrName³

Descrizione: Contiene il nome del gestore code locale dopo che è stata eseguita la risoluzione del nome.

Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_MGR
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_MQ_Q_MGR_

ReportOptions

Descrizione: Opzioni di report dei messaggi
Parametro PCF: REPORT MQIACF
Livello di traccia: 2
Tipo: MQCFIN

MsgType

Descrizione: Tipo di messaggio
Parametro PCF: TIPO_MSG_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

Expiry

Descrizione: Durata messaggio
Parametro PCF: MQIACF_SCADENZA
Livello di traccia: 2
Tipo: MQCFIN

Format

Descrizione: Nome formato dei dati di messaggio
Parametro PCF: MQCACH_FORMAT_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: MQ_FORMAT_LENGTH

Priority

Descrizione: Priorità messaggio
Parametro PCF: PRIORITà_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

Persistence

Descrizione: Persistenza messaggio
Parametro PCF: MQIACF_PERSISTENZA
Livello di traccia: 2
Tipo: MQCFIN

MsgId

Descrizione: ID messaggio
Parametro PCF: ID_MSG_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: ID_MQ_MSG_LENGTH

CorrelId

Descrizione: Identificativo di correlazione
Parametro PCF: ID_CORREL_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: LUNGHEZZA_ID_CORRELATA

ReplyToQueue

Descrizione:
Parametro PCF: MQCACF_REPLY_TO_Q
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

ReplyToQMgr

Descrizione:
Parametro PCF: MQCACF_REPLY_TO_Q_MGR
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_MQ_Q_MGR_

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio
Parametro PCF: ID_MQIA_CODED_CHAR_SET_
Livello di traccia: 2
Tipo: MQCFIN

Encoding

Descrizione: Codifica numerica dei dati del messaggio.
Parametro PCF: MQIACF_ENCODING
Livello di traccia: 2
Tipo: MQCFIN

PutDate

Descrizione:
Parametro PCF: DATA_PUT_MQCACF
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_DATA_PUT_MQ

PutTime

Descrizione:
Parametro PCF: ORA_PUT_MQCACF
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_TEMPO_PUT_MQ

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione: Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: La lunghezza varia.

ResolvedType

Descrizione: Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF: TIPO_RESOLV_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

PolicyName

Descrizione: Il nome della politica applicato a questo messaggio.
Nota: Solo messaggi protetti AMS
Parametro PCF: NOME_POLITICA_MQCA
Livello di traccia: 2

Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_OGGETTO_MQ

XmitqMsgId

Descrizione: L'ID del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: ID_MSG_XQH_MQBACF

Livello di traccia: 2

Tipo: MQCFB

Lunghezza: ID_MQ_MSG_LENGTH

XmitqCorrelId

Descrizione: L'ID di correlazione del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQBACF_XQH_CORREL_ID

Livello di traccia: 2

Tipo: MQCFB

Lunghezza: LUNGHEZZA_ID_CORRELATA

XmitqPutTime

Descrizione: L'ora di immissione del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_PUT_TIME

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_TEMPO_PUT_MQ

XmitqPutDate

Descrizione: La data di inserimento del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_PUT_DATE

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_DATA_PUT_MQ

XmitqRemoteQName

Descrizione: La destinazione della coda remota del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_REMOTE_Q_NAME

Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

XmitqRemoteQMgr

Descrizione: La destinazione del gestore code remoto del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQCACF_XQH_REMOTE_Q_MGR
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

PutMsgOptsStructure

Descrizione: La struttura MQPMO.
Parametro PCF: MQBACF_MQPMO_STRUCTURE
Livello di traccia: 3
Tipo: MQCFB
Lunghezza: La lunghezza in byte della struttura MQPMO (la dimensione effettiva dipende dalla versione della struttura)

QMgrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION
Livello di traccia: 2
Tipo: MQCFIN64

Struttura intestazione gruppo PCF elenco di distribuzione attività applicazione MQPUT

Se la funzione MQPUT sta inserendo in un elenco di distribuzione, i parametri MQPUT includono un gruppo PCF AppActivityDistList . Per ogni coda nell'elenco di distribuzione, consultare [“Struttura intestazione gruppo PCF elenco di distribuzione attività applicazione” a pagina 260](#). Il gruppo PCF AppActivityDistList combina le informazioni dalle strutture MQPMR e MQRR per identificare i parametri PUT e indicare il risultato dell'operazione PUT su ciascuna coda. Per le operazioni MQPUT il gruppo AppActivityDistList contiene alcuni o tutti i seguenti parametri (CompCode e Reason sono presenti se il codice di errore è MQRC_MULTIPLE_REASON e gli altri parametri sono determinati da MQPMO.PutMsgRecFields):

³ Il parametro QName ResolvedLocal è incluso solo se è diverso dal parametro ResolvedQName .

CompCode

Descrizione:	Il codice di completamento che indica il risultato dell'operazione. Incluso solo se vengono fornite le strutture MQRR e il codice motivo per MQPUT è MQRC_MULTIPLE_REASON
Parametro PCF:	CODICE_COMP_MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

Reason

Descrizione:	Il codice motivo che indica il risultato dell'inserimento per questo oggetto. Incluso solo se vengono fornite le strutture MQRR e il codice motivo per MQPUT è MQRC_MULTIPLE_REASON
Parametro PCF:	CODICE_REASON_MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

MsgId

Descrizione:	L'identificativo del messaggio. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_MSG_ID
Parametro PCF:	ID_MSG_MQBACF
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza:	ID_MQ_MSG_LENGTH

CorrelId

Descrizione:	Identificativo di correlazione. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_CORREL_ID
Parametro PCF:	ID_CORREL_MQBACF
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza:	LUNGHEZZA_ID_CORRELATA

GroupId

Descrizione:	Identificativo gruppo. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_GROUP_ID
Parametro PCF:	ID_GROUP_MQBACF
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza:	LUNGHEZZA_ID_GRUPPO_MQ.

Feedback

Descrizione:	Feedback. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_FEEDBACK
--------------	--

Parametro PCF: MQIACF_FEEDBACK
Livello di traccia: 2
Tipo: MQCFIN

AccountingToken

Descrizione: AccountingToken. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_ACCOUNTING_TOKEN
Parametro PCF: MQBACF_ACCOUNTING_TOKEN
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: MQ_ACCOUNTING_TOKEN_LENGTH.

MQPUT1

L'applicazione ha avviato la funzione MQI MQPUT1

ObjectType

Descrizione: Il tipo di oggetto passato in MQOT.ObjectType
Parametro PCF: TIPO_OGGETTO_MQIAC
Livello di traccia: 1
Tipo: MQCFIN

ObjectName

Descrizione: Il nome dell'oggetto passato alla chiamata MQI prima che venga tentata qualsiasi risoluzione del nome della coda.
Parametro PCF: MQCACF_XX_ENCODE_CASE_ONE nome_oggetto
Livello di traccia: 1
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

ObjectQMgrName

Descrizione: Il nome del gestore code oggetti passato alla chiamata MQI prima che venga tentata qualsiasi risoluzione del nome coda.
Parametro PCF: MQCACF_OBJECT_Q_MGR_NAME
Livello di traccia: 1
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_MQ_Q_MGR_

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

PutOptions

Descrizione: Le opzioni put da MQPMO.Options
Parametro PCF: OPZIONI_MQIACF_PUT_
Livello di traccia: 1
Tipo: MQCFIN

AlternateUserId

Descrizione: Incluso solo se MQPMO_ALTERNATE_USER_AUTHORITY è specificato.
Parametro PCF: MQCACF_ALTERNATE_USERID
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_ID_UTENTE_MQ

RecsPresent

Descrizione: Il numero di record nome oggetto presenti
Parametro PCF: MQIACF_RECS_PRESENTE
Livello di traccia: 1
Tipo: MQCFIN

KnownDestCount

Descrizione: Numero di code locali aperte correttamente
Parametro PCF: CONTEGGIO_DEST_CONOSCIUTO_MQIACF_
Livello di traccia: 1
Tipo: MQCFIN

UnknownDestCount

Descrizione: Numero di code remote aperte correttamente
Parametro PCF: MQIACF_UNKNOWN_DEST_COUNT
Livello di traccia: 1
Tipo: MQCFIN

InvalidDestCount

Descrizione: Numero di code che non è stato possibile aprire
Parametro PCF: MQIACF_INVALID_DEST_COUNT
Livello di traccia: 1

Tipo: MQCFIN

MsgBuffer

Descrizione: Dati del messaggio.

Parametro PCF: DATI_MESSAGE_MQBACF

Livello di traccia: 1

Tipo: MQCFB

Lunghezza: La lunghezza è regolata dal parametro TRACEDATA () impostato nella configurazione APPTRACE. Se TRACEDATA=NONE, questo parametro viene omissso.

MsgLength

Descrizione: Lunghezza del messaggio.

Parametro PCF: MQIACF_MSG_LENGTH

Livello di traccia: 1

Tipo: MQCFIN

HighResTime

Descrizione: Ora dell'operazione in microsecondi dalla mezzanotte del 1 gennaio 1st 1970 (UTC)

Nota: L'accuratezza di questo timer varia in base al supporto della piattaforma per un timer ad alta risoluzione.

Parametro PCF: MQIAMO64_HIGHRES_TIME

Livello di traccia: 2

Tipo: MQCFIN64

ResolvedQName

Descrizione: Il nome della coda dopo che è stata eseguita la risoluzione del nome della coda.

Parametro PCF: MQCACF_RESOLVED_Q_NAME

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_Q_MQ

ResolvedQMgrName

Descrizione: Il nome del gestore code dopo che è stata eseguita la risoluzione del nome.

Parametro PCF: MQCACF_RESOLVED_Q_MGR

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_MQ_Q_MGR_

ResolvedLocalQName ⁴

Descrizione: Contiene il nome della coda locale dopo che è stata eseguita la risoluzione del nome

Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia: 2
Tipo: MQCFST

ResolvedLocalQMgrName ⁴

Descrizione: Contiene il nome del gestore code locale dopo che è stata eseguita la risoluzione del nome.
Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_MGR
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_MQ_Q_MGR_

AlternateSecurityId

Descrizione: Identificativo di sicurezza alternativo. Presente solo se MQOD.Version è uguale o superiore a MQOD.VERSION_3 e MQOD.AlternateSecurityId non è uguale a MQSID_NONE.
Parametro PCF: ID_SICUREZZA_ALTERNATO_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: LUNGHEZZA_ID_SICUREZZA_MQ

ObjectString

Descrizione: Nome oggetto lungo. Incluso solo se MQOD.Version è uguale o superiore a MQOD.VERSION_4 e al campo VSLength di MQOD.ObjectString è MQVS_NULL_TERMINATED o è maggiore di zero.
Parametro PCF: MQCACF_STRINGA_OGGETTO
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: La lunghezza varia.

ResObjectString

Descrizione: Il nome oggetto lungo dopo che il gestore code ha risolto il nome fornito nel campo ObjectName . Incluso solo per gli argomenti e gli alias della coda che fanno riferimento a un oggetto argomento se MQOD.Version è uguale o superiore a MQOD.VERSION_4 e VSLength è MQVS_NULL_TERMINATED o superiore a zero.
Parametro PCF: MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: La lunghezza varia.

ResolvedType

Descrizione:	Il tipo di oggetto risolto (base) che si sta aprendo. Incluso solo se MQOD.Version è uguale o superiore a MQOD_VERSION_4. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF:	TIPO_RESOLV_MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

ReportOptions

Descrizione:	Opzioni di report dei messaggi
Parametro PCF:	REPORT MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

MsgType

Descrizione:	Tipo di messaggio
Parametro PCF:	TIPO_MSG_MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

Expiry

Descrizione:	Durata messaggio
Parametro PCF:	MQIACF_SCADENZA
Livello di traccia:	2
Tipo:	MQCFIN

Format

Descrizione:	Nome formato dei dati di messaggio
Parametro PCF:	MQCACH_FORMAT_NAME
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	MQ_FORMAT_LENGTH

Priority

Descrizione:	Priorità messaggio
Parametro PCF:	PRIORITÀ_MQIACF
Livello di traccia:	2
Tipo:	MQCFIN

Persistence

Descrizione:	Persistenza messaggio
Parametro PCF:	MQIACF_PERSISTENZA

Livello di traccia: 2
Tipo: MQCFIN

MsgId

Descrizione: ID messaggio
Parametro PCF: ID_MSG_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: ID_MQ_MSG_LENGTH

CorrelId

Parametro PCF: Identificativo di correlazione
Descrizione: ID_CORREL_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: LUNGHEZZA_ID_CORRELATA

ReplyToQueue

Descrizione:
Parametro PCF: MQCACF_REPLY_TO_Q
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

ReplyToQMgr

Descrizione:
Parametro PCF: MQCACF_REPLY_TO_Q_MGR
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: MQCFST

CodedCharSetId

Descrizione: CSID (Character set identifier) dei dati del messaggio
Parametro PCF: ID_MQIA_CODED_CHAR_SET_
Livello di traccia: 2
Tipo: MQCFIN

Encoding

Descrizione: Codifica numerica dei dati del messaggio.
Parametro PCF: MQIACF_ENCODING
Livello di traccia: 2

Tipo: MQCFIN

PutDate

Descrizione:

Parametro PCF: DATA_PUT_MQCACF

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_DATA_PUT_MQ

PutTime

Descrizione:

Parametro PCF: ORA_PUT_MQCACF

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_TEMPO_PUT_MQ

PolicyName

Descrizione: Il nome della politica applicato a questo messaggio.

Nota: Solo messaggi protetti AMS

Parametro PCF: NOME_POLITICA_MQCA

Livello di traccia: 2

Tipo: MQCFST

Lunghezza: LUNGHEZZA_NOME_OGGETTO_MQ

XmitqMsgId

Descrizione: L'ID del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: ID_MSG_XQH_MQBACF

Livello di traccia: 2

Tipo: MQCFB

Lunghezza: ID_MQ_MSG_LENGTH

XmitqCorrelId

Descrizione: L'ID di correlazione del messaggio nell'intestazione della coda di trasmissione.

Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER

Parametro PCF: MQBACF_XQH_CORREL_ID

Livello di traccia: 2

Tipo: MQCFB

Lunghezza: LUNGHEZZA_ID_CORRELATA

XmitqPutTime

Descrizione:	L'ora di immissione del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_PUT_TIME
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_TEMPO_PUT_MQ

XmitqPutDate

Descrizione:	La data di inserimento del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_PUT_DATE
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_DATA_PUT_MQ

XmitqRemoteQName

Descrizione:	La destinazione della coda remota del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_REMOTE_Q_NAME
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_NOME_Q_MQ

XmitqRemoteQMgr

Descrizione:	La destinazione del gestore code remoto del messaggio nell'intestazione della coda di trasmissione. Nota: Solo quando il formato è MQFMT_XMIT_Q_HEADER
Parametro PCF:	MQCACF_XQH_REMOTE_Q_MGR
Livello di traccia:	2
Tipo:	MQCFST
Lunghezza:	LUNGHEZZA_NOME_Q_MQ

PutMsgOptsStructure

Descrizione:	La struttura MQPMO.
Parametro PCF:	MQBACF_MQPMO_STRUCT
Livello di traccia:	3
Tipo:	MQCFB

Lunghezza: La lunghezza in byte della struttura MQPMO (la dimensione effettiva dipende dalla versione della struttura)

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo: MQCFIN64

MQPUT1 AppActivityDistList Struttura intestazione gruppo PCF

Se la funzione MQPUT1 sta inserendo un elenco di distribuzione, i parametri della variabile includono un gruppo PCF AppActivityDistList . Per ogni coda nell'elenco di distribuzione, consultare “Struttura intestazione gruppo PCF elenco di distribuzione attività applicazione” a pagina 260. Il gruppo PCF AppActivityDistList combina le informazioni dalle strutture MQOR, MQPMR e MQRR per identificare gli oggetti e i parametri PUT e indicare il risultato dell'operazione PUT su ciascuna coda. Per operazioni MQPUT1 il gruppo AppActivityDistList contiene alcuni o tutti i seguenti parametri (CompCode, Reason, ObjectName e ObjectQMGrName è presente se il codice di errore è MQRC_MULTIPLE_REASON e gli altri parametri sono determinati da MQPMO.PutMsgRecFields):

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'inserimento per questo oggetto. Incluso solo se vengono fornite strutture MQRR e il codice motivo per MQPUT1 è MQRC_MULTIPLE_REASON

Parametro PCF: CODICE_COMP_MQIACF

Livello di traccia: 2

Tipo: MQCFIN

Reason

Descrizione: Il codice motivo che indica il risultato dell'inserimento per questo oggetto. Incluso solo se vengono fornite strutture MQRR e il codice motivo per MQPUT1 è MQRC_MULTIPLE_REASON

Parametro PCF: CODICE_REASON_MQIACF

Livello di traccia: 2

Tipo: MQCFIN

ObjectName

Descrizione: Il nome di una coda nell'elenco di distribuzione. Incluso solo se vengono fornite le strutture MQOR.

Parametro PCF: MQCACF_XX_ENCODE_CASE_ONE nome_oggetto

⁴ Il parametro QName ResolvedLocal è incluso solo se è diverso dal parametro ResolvedQName .

Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

MsgId

Descrizione: L'identificativo del messaggio. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_MSG_ID
Parametro PCF: ID_MSG_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: ID_MQ_MSG_LENGTH

CorrelId

Descrizione: Identificativo di correlazione. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_CORREL_ID
Parametro PCF: ID_CORREL_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: LUNGHEZZA_ID_CORRELATA

GroupId

Descrizione: Identificativo gruppo. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_GROUP_ID
Parametro PCF: ID_GROUP_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: LUNGHEZZA_ID_GRUPPO_MQ.

Feedback

Descrizione: Feedback. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_FEEDBACK
Parametro PCF: MQIACF_FEEDBACK
Livello di traccia: 2
Tipo: MQCFIN

AccountingToken

Descrizione: AccountingToken. Incluso solo se le strutture MQPMR sono provided.and PutMsgRecFields include MQPMRF_ACCOUNTING_TOKEN
Parametro PCF: MQBACF_ACCOUNTING_TOKEN
Livello di traccia: 2
Tipo: MQCFB
Lunghezza: MQ_ACCOUNTING_TOKEN_LENGTH.

MQSET

L'applicazione ha avviato la funzione MQSET MQI

ObjectHandle

Descrizione: L'handle dell'oggetto
Parametro PCF: HOBJ MQIACF
Livello di traccia: 1
Tipo: MQCFIN

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione
Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

SelectorCount

Descrizione: Il numero di selettori forniti nell'array Selettori.
Parametro PCF: MQIACF_SELECTOR_COUNT
Livello di traccia: 2
Tipo: MQCFIN

Selectors

Descrizione: L'elenco di attributi (numero intero o carattere) i cui valori vengono aggiornati da MQSET.
Parametro PCF: SELECTORS MQIACF
Livello di traccia: 2
Tipo: MQCFIL

ResolvedQName

Descrizione: Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF: MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia: 2
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione:	Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF:	MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	La lunghezza varia.

ResolvedType

Descrizione:	Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF:	TIPO_RESOLV_MQIACF
Livello di traccia:	2
Tipo	MQCFIN

IntAttrCount

Descrizione:	Il numero di attributi interi che devono essere aggiornati dall'operazione di impostazione.
Parametro PCF:	Conteggio_INTATTR_MQIACF_
Livello di traccia:	3
Tipo:	MQCFIN

IntAttr

Descrizione:	I valori dell'attributo integer
Parametro PCF:	MQIACF_INT_ATTRS
Livello di traccia:	3
Tipo:	MQCFIL
Intervallo:	Questo parametro è presente solo se il conteggio IntAttrè > 0

CharAttr

Descrizione:	Gli attributi del carattere che devono essere aggiornati dall'operazione di impostazione. I valori sono concatenati insieme.
Parametro PCF:	MQCACF_CHAR_ATTRS
Livello di traccia:	3
Tipo:	MQCFST
Intervallo:	Questo parametro è incluso solo se la lunghezza CharAttrè > 0

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION

Livello di traccia: 2

Tipo: MQCFIN64

MQSUB

L'applicazione ha avviato la funzione MQSUB MQI

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione

Parametro PCF: CODICE_COMP_MQIACF

Livello di traccia: 1

Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione

Parametro PCF: CODICE_REASON_MQIACF

Livello di traccia: 1

Tipo: MQCFIN

SubHandle

Descrizione: L'handle di sottoscrizione

Parametro PCF: HSUB MQIACF

Livello di traccia: 1

Tipo: MQCFIN

ObjectHandle

Descrizione: L'handle dell'oggetto

Parametro PCF: HOBJ MQIACF

Livello di traccia: 1

Tipo: MQCFIN

Options

Descrizione: Opzioni di sottoscrizione

Parametro PCF: MQIACF_SUB_OPZIONI

Livello di traccia: 1

Tipo: MQCFIN

ObjectName

Descrizione: Il nome dell'oggetto.
Parametro PCF: MQCACF_XX_ENCODE_CASE_ONE nome_oggetto
Livello di traccia: 1
Tipo: MQCFST
Lunghezza: LUNGHEZZA_NOME_Q_MQ

ObjectString

Descrizione: Nome oggetto lungo.
Parametro PCF: MQCACF_STRINGA_OGGETTO
Livello di traccia: 1
Tipo: MQCFST
Intervallo: Incluso solo se il campo VSLength di MQSD.ObjectString è maggiore di zero o MQVS_NULL_TERMINATED.
Lunghezza: La lunghezza varia.

AlternateUserId

Descrizione:
Parametro PCF: MQCACF_ALTERNATE_USERID
Livello di traccia: 2
Tipo: MQCFST
Intervallo: Incluso solo se è specificato MQSO_ALTERNATE_USER_AUTHORITY.
Lunghezza: LUNGHEZZA_ID_UTENTE_MQ

AlternateSecurityId

Descrizione: Identificativo di sicurezza alternativo.
Parametro PCF: ID_SICUREZZA_ALTERNATO_MQBACF
Livello di traccia: 2
Tipo: MQCFB
Intervallo: Presente solo se è stato specificato MQSO_ALTERNATE_USER_AUTHORITY e MQSD.AlternateSecurityId non è uguale a MQSID_NONE.
Lunghezza: LUNGHEZZA_ID_SICUREZZA_MQ

SubName

Descrizione: Nome sottoscrizione
Parametro PCF: MQCACF_SUB_NAME
Livello di traccia: 2
Tipo: MQCFST
Intervallo: Incluso solo se il campo VSLength di MQSD.SubName è maggiore di zero o MQVS_NULL_TERMINATED.
Lunghezza: La lunghezza varia.

SubUserData

Descrizione:	Dati utente sottoscrizione
Parametro PCF:	DATI_USER_MQCACF
Livello di traccia:	2
Tipo:	MQCFST
Intervallo:	Incluso solo se il campo VSLength di MQSD.SubName è maggiore di zero o MQVS_NULL_TERMINATED.
Lunghezza:	La lunghezza varia.

SubCorrelId

Descrizione:	Identificativo di correlazione sottoscrizione
Parametro PCF:	ID_CORREL_MQBACF
Livello di traccia:	2
Tipo:	MQCFB
Lunghezza:	LUNGHEZZA_ID_CORRELATA

SelectionString

Descrizione:	Stringa di selezione.
Parametro PCF:	MQCACF_SELECTION_STRING
Livello di traccia:	2
Tipo:	MQCFST
Intervallo:	Incluso solo se il campo VSLength di MQSD. SelectionString è MQVS_NULL_TERMINATED o maggiore di zero.
Lunghezza:	La lunghezza varia.

ResolvedQName

Descrizione:	Il nome coda a cui fa riferimento ObjectHandle, quando ResolvedType è MQOT_Q.
Parametro PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	LUNGHEZZA_NOME_Q_MQ.

ResObjectString

Descrizione:	Il nome oggetto a cui fa riferimento l' ObjectHandle, quando ResolvedType è MQOT_TOPIC.
Parametro PCF:	MQCACF_RESOLVED_OBJECT_STRING
Livello di traccia:	2
Tipo	MQCFST
Lunghezza:	La lunghezza varia.

ResolvedType

Descrizione:	Il tipo di oggetto a cui fa riferimento ObjectHandle. I valori possibili sono MQOT_Q, MQOT_TOPIC o MQOT_NONE.
Parametro PCF:	TIPO_RESOLV_MQIACF
Livello di traccia:	2
Tipo	MQCFIN

SubDescriptorStructure

Descrizione:	La struttura MQSD.
Parametro PCF:	MQBACF_MQSD_STRUCT
Livello di traccia:	3
Tipo:	MQCFB
Lunghezza:	La lunghezza, in byte, della struttura MQSD.

QMGrOpDuration

Descrizione:	Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code. La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ . Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.
Parametro PCF:	MQIAMO64_QMGR_OP_DURATION
Livello di traccia:	2
Tipo	MQCFIN64

MQSUBRQ

L'applicazione ha avviato la funzione MQI MQSUBRQ

CompCode

Descrizione:	Il codice di completamento che indica il risultato dell'operazione
Parametro PCF:	CODICE_COMP_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

Reason

Descrizione:	Il risultato del codice motivo dell'operazione
Parametro PCF:	CODICE_REASON_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

SubHandle

Descrizione:	L'handle di sottoscrizione
Parametro PCF:	HSUB MQIACF

Livello di traccia: 1
Tipo: MQCFIN

SubOptions

Descrizione: Le opzioni secondarie da MQSB.Options
Parametro PCF: MQIACF_SUBRQ_OPZIONI
Livello di traccia: 2
Tipo: MQCFIN

Action

Descrizione: L'azione di richiesta di sottoscrizione (MQSR_*)
Parametro PCF: AZIONE MQIACF_SUBRQ_
Livello di traccia: 2
Tipo: MQCFIN

NumPubs

Descrizione: Il numero di pubblicazioni inviate come risultato di questa chiamata (da MQSB.NumPubs)
Parametro PCF: NUM_PUBS MQIACF
Livello di traccia: 2
Tipo: MQCFIN

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION
Livello di traccia: 2
Tipo: MQCFIN64

MQSTAT

L'applicazione ha avviato la funzione MQI MQSTAT

CompCode

Descrizione: Il codice di completamento che indica il risultato dell'operazione
Parametro PCF: CODICE_COMP_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Reason

Descrizione: Il risultato del codice motivo dell'operazione

Parametro PCF: CODICE_REASON_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Type

Descrizione: Tipo di informazioni di stato richieste
Parametro PCF: TIPO_STATO_MQIACF
Livello di traccia: 2
Tipo: MQCFIN

StatusStructure

Descrizione: La struttura MQSTS.
Parametro PCF: MQBACF_MQSTS_STRUCT
Livello di traccia: 3
Tipo: MQCFB
Lunghezza: La lunghezza in byte della struttura MQSTS (la dimensione effettiva dipende dalla versione della struttura)

QMGrOpDuration

Descrizione: Durata approssimativa della chiamata API, in microsecondi, all'interno del gestore code.

La durata non include il tempo trascorso all'esterno del gestore code. Ad esempio, il tempo impiegato come client IBM MQ .

Nota: L'accuratezza di questo timer varia a seconda della piattaforma utilizzata dall'azienda.

Parametro PCF: MQIAMO64_QMGR_OP_DURATION
Livello di traccia: 2
Tipo: MQCFIN64

Parametri variabili per le operazioni XA dell'attività dell'applicazione

Le operazioni XA sono chiamate API che le applicazioni possono effettuare per abilitare MQ a partecipare a una transazione. I parametri per ciascuna operazione sono definiti nella seguente sezione.

Il livello di traccia indica il livello di granularità della traccia richiesto per i parametri da includere nella traccia. I valori possibili del livello di traccia sono:

1. Bassa

Il parametro viene incluso quando la traccia di attività "bassa", "media" o "alta" è configurata per un'applicazione. Questa impostazione indica che un parametro è sempre incluso nel gruppo AppActivityData per l'operazione. Questa serie di parametri è sufficiente per tenere traccia delle chiamate MQI effettuate da un'applicazione e per verificare se hanno esito positivo.

2. Medio

Il parametro è incluso nel gruppo AppActivityData per l'operazione solo quando la traccia "media" o "alta" attività è configurata per un'applicazione. Questa serie di parametri aggiunge informazioni sulle risorse, ad esempio, i nomi di coda e argomento utilizzati dall'applicazione.

3. Elevato

Il parametro è incluso solo nel gruppo `AppActivityData` per l'operazione quando la traccia "alta" attività è configurata per un'applicazione. Questa serie di parametri include i dump di memoria delle strutture trasmesse alle funzioni MQI e XA. Per questo motivo, contiene ulteriori informazioni sui parametri utilizzati nelle chiamate MQI e XA. I dump di memoria della struttura sono copie poco profonde delle strutture. Per evitare tentativi errati di annullare il riferimento dei puntatori, i valori del puntatore nelle strutture sono impostati su NULL.

Nota: La versione della struttura di cui viene eseguito il dump non è necessariamente identica alla versione utilizzata da una applicazione. La struttura può essere modificata da un'uscita incrociata API, dal codice di traccia dell'attività o dal gestore code. Un gestore code può modificare una struttura in una versione successiva, ma il gestore code non la modifica mai in una versione precedente della struttura. Per farlo, rischierebbe di perdere dati.

AXREG

L'applicazione ha avviato la funzione AXREG AX

XID

Descrizione:	La struttura XID
Parametro PCF:	XID_XA_MQBACF
Livello di traccia:	1
Tipo:	MQCFB
Lunghezza:	Dimensione (XID)

Rmid

Descrizione:	Identificativo gestore risorse
Parametro PCF:	IDRM_XA_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

Flags

Descrizione:	Indicatori
Parametro PCF:	MQIACF_XA_FLAGS
Livello di traccia:	1
Tipo:	MQCFIN

XARetCode

Descrizione:	Codice di ritorno
Parametro PCF:	CODICE_XA_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

AXUNREG

L'applicazione ha avviato la funzione AXUNREG AX

Rmid

Descrizione:	Identificativo gestore risorse
Parametro PCF:	IDRM_XA_MQIACF
Livello di traccia:	1

Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XACLOSE

L'applicazione ha avviato la funzione XACLOSE AX

Xa_info

Descrizione: Informazioni utilizzate per inizializzare il gestore risorse.
Parametro PCF: INFO_XA_MQCACF
Livello di traccia: 1
Tipo: MQCFST

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

COMMIT

L'applicazione ha avviato la funzione XACOMMIT AX

XID

Descrizione: La struttura XID
Parametro PCF: XID_XA_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: Dimensione (XID)

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XACOMPLETE

L'applicazione ha avviato la funzione XACOMPLETE AX

Handle

Descrizione: Handle per operazione asincrona
Parametro PCF: MQIACF_XA_HANDLE
Livello di traccia: 1
Tipo: MQCFIN

Retval

Descrizione: Valore di ritorno della funzione asincrona
Parametro PCF: XA_RETVAL MQIACF_AL
Livello di traccia: 1
Tipo: MQCFINMQCFBS

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF

Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

INVIA

L'applicazione ha avviato la funzione XAEND AX

XID

Descrizione: La struttura XID
Parametro PCF: XID_XA_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: Dimensione (XID)

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XAFORGET

L'applicazione ha avviato la funzione AXREG AX

XID

Descrizione:	La struttura XID
Parametro PCF:	XID_XA_MQBACF
Livello di traccia:	1
Tipo:	MQCFB
Lunghezza:	Dimensione (XID)

Rmid

Descrizione:	Identificativo gestore risorse
Parametro PCF:	IDRM_XA_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

Flags

Descrizione:	Indicatori
Parametro PCF:	MQIACF_XA_FLAGS
Livello di traccia:	1
Tipo:	MQCFIN

XARetCode

Descrizione:	Codice di ritorno
Parametro PCF:	CODICE_XA_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

XAOPEN

L'applicazione ha avviato la funzione XAOPEN AX

Xa_info

Descrizione:	Informazioni utilizzate per inizializzare il gestore risorse.
Parametro PCF:	INFO_XA_MQCACF
Livello di traccia:	1
Tipo:	MQCFST

Rmid

Descrizione:	Identificativo gestore risorse
Parametro PCF:	IDRM_XA_MQIACF
Livello di traccia:	1
Tipo:	MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XAPREPARE

L'applicazione ha avviato la funzione XAPREPARE AX

XID

Descrizione: La struttura XID
Parametro PCF: XID_XA_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: Dimensione (XID)

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XARECOVER

L'applicazione ha avviato la funzione XARECOVER AX

Count

Descrizione: Numero di XID
Parametro PCF: CONTO_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XIDs

Descrizione: Le strutture XID
Nota: Ci sono più istanze di questo parametro PCF - una per ogni struttura XID fino a Conteggio XID
Parametro PCF: XID_XA_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: Dimensione (XID)

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XAROLLBACK

L'applicazione ha avviato la funzione XAROLLBACK AX

XID

Descrizione: La struttura XID
Parametro PCF: XID_XA_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: Dimensione (XID)

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

XASTART

L'applicazione ha avviato la funzione XASTART AX

XID

Descrizione: La struttura XID
Parametro PCF: XID_XA_MQBACF
Livello di traccia: 1
Tipo: MQCFB
Lunghezza: Dimensione (XID)

Rmid

Descrizione: Identificativo gestore risorse
Parametro PCF: IDRM_XA_MQIACF
Livello di traccia: 1
Tipo: MQCFIN

Flags

Descrizione: Indicatori
Parametro PCF: MQIACF_XA_FLAGS
Livello di traccia: 1
Tipo: MQCFIN

XARetCode

Descrizione: Codice di ritorno
Parametro PCF: CODICE_XA_MQIACF

Livello di traccia: 1
Tipo: MQCFIN

Multi Argomenti di sistema per il monitoraggio e la traccia dell'attività

Gli argomenti di sistema nelle strutture ad albero degli argomenti del gestore code vengono utilizzati per il controllo delle risorse (alcuni dei quali sono simili al contenuto dei messaggi delle statistiche) e come un modo per utilizzare la traccia dell'attività dell'applicazione.

Ramo \$SYS/MQ della struttura ad albero degli argomenti del gestore code

Ogni struttura di argomenti del gestore code contiene il ramo \$SYS/MQ . Il gestore code pubblica le stringhe argomento in questo ramo. Un utente autorizzato può sottoscrivere queste stringhe argomento per ricevere informazioni sul gestore code e sull'attività su di esso. Questi argomenti di sistema vengono utilizzati per la traccia e il controllo dell'attività dell'applicazione. Per ulteriori informazioni, consultare [Strutture ad albero degli argomenti](#).

La root del ramo \$SYS/MQ è rappresentata da SYSTEM.ADMIN.TOPIC . Il ramo \$SYS/MQ della struttura ad albero degli argomenti è isolato dal resto della struttura ad albero degli argomenti nei modi seguenti:

- Una sottoscrizione effettuata con caratteri jolly in un punto più alto della struttura ad albero di \$SYS/MQ non corrisponde ad alcuna stringa di argomenti all'interno del ramo \$SYS/MQ . L'operazione con caratteri jolly per SYSTEM.ADMIN.TOPIC è impostato su "Blocco" e non può essere modificato. Questa limitazione si applica anche quando si utilizzano caratteri jolly con il comando **runmqsc DISPLAY TPSTATUS** per visualizzare i nodi nella struttura ad albero degli argomenti. Per visualizzare i nodi argomento all'interno del ramo \$SYS/MQ , avviare la stringa argomento con \$SYS/MQ . Ad esempio, utilizzare \$SYS/MQ/# per visualizzare tutti i nodi.
- È necessario essere autorizzati a \$SYS/MQ o più in profondità per poter utilizzare la struttura ad albero degli argomenti \$SYS/MQ . L'autorizzazione a sottoscrivere una stringa di argomenti si basa sull'autorizzazione concessa per un oggetto argomento gestito uguale o superiore alla stringa di argomenti nella struttura ad albero degli argomenti. Le autorizzazioni concesse alla root (SYSTEM.BASE.TOPIC) concederebbe un'autorità utente a tutte le stringhe di argomento. Tuttavia, nel caso del ramo \$SYS/MQ , l'accesso concesso superiore a \$SYS/MQ non si applica alle stringhe dell'argomento \$SYS/MQ .
- Il ramo \$SYS/MQ della struttura ad albero dell'argomento è isolato dagli attributi dell'argomento impostati più in alto nella struttura ad albero. Il SISTEMA SYSTEM.ADMIN.TOPIC non eredita alcun attributo da un oggetto argomento definito più in alto nella struttura di argomenti. Ad esempio, la modifica degli attributi di SYSTEM.BASE.TOPIC non influisce sulla funzionalità del ramo \$SYS/MQ .

Tutte le stringhe argomento che iniziano con \$SYS/MQ sono riservate per l'utilizzo da parte di IBM MQ. Queste stringhe di argomenti hanno le seguenti restrizioni:

- Non è possibile abilitare multicast dal ramo \$SYS/MQ della struttura ad albero degli argomenti.
- Il clustering non è supportato per il ramo \$SYS/MQ .
- Il dispositivo di sottoscrizione proxy non può essere impostato su "force".
- Le applicazioni non possono pubblicare una stringa di argomenti \$SYS/MQ .
- Per impostazione predefinita, l'ambito di pubblicazione e sottoscrizione è solo il gestore code locale.
- Non è possibile utilizzare caratteri jolly nei punti seguenti:
 - \$SYS/MQ/
 - \$SYS/MQ/INFO
 - \$SYS/MQ/INFO/QMGR
 - \$SYS/MQ/INFO/QMGR/nome_gestore_code
 - \$SYS/MQ/INFO/QMGR/nome_gestore_code/ActivityTrace

I tentativi di utilizzare i caratteri jolly in questi punti causano un errore di sottoscrizione con il motivo MQRC_ADMIN_TOPIC_STRING_ERROR.

API di monitoraggio delle prestazioni disponibile tramite pubblicazione / sottoscrizione

È possibile monitorare le statistiche del gestore code utilizzando il meccanismo di pubblicazione / sottoscrizione semplice e l'applicazione di esempio amqsrua o la propria applicazione. Le statistiche vengono pubblicate in un argomento di sistema in \$SYS/MQ/INFO/QMGR che consente agli utenti di monitorare le risorse. Queste statistiche possono essere visualizzate eseguendo l'applicazione di esempio amqsrua o scrivendo un'applicazione che sottoscrive l'argomento del sistema di monitoraggio delle risorse in modo simile a **amqsrua**.

Concetti correlati

[“Traccia attività applicazione” a pagina 207](#)

La traccia dell'attività dell'applicazione produce informazioni dettagliate sul comportamento delle applicazioni connesse a un gestore code. Traccia il comportamento di una applicazione e fornisce una vista dettagliata dei parametri utilizzati da un'applicazione quando interagisce con risorse IBM MQ . Inoltre, mostra la sequenza di chiamate MQI emesse da una applicazione.

Monitoraggio dell'utilizzo delle risorse del sistema utilizzando il comando amqsrua

È possibile utilizzare il comando **amqsrua** per interrogare i dati sulle prestazioni relativi all'utilizzo delle risorse di sistema di un gestore code.

Informazioni su questa attività

L'applicazione di esempio amqsrua mostra un modo per utilizzare le pubblicazioni di monitoraggio IBM MQ e visualizzare i dati sulle prestazioni pubblicati dai gestori code. Questi dati possono includere informazioni sulla CPU, la memoria e l'utilizzo del disco. È inoltre possibile visualizzare dati equivalenti ai dati delle statistiche PCF STATMQI. I dati vengono pubblicati ogni 10 secondi e vengono riportati durante l'esecuzione del comando.

È possibile eseguire il comando solo con il nome del gestore code e procedere in modo interattivo per scegliere i parametri **CLASS**, quindi **TYPE** e quindi **object** disponibili per il gestore code ad ogni passo. Se si conoscono i nomi **CLASS**, **TYPE** e **object** per i quali si desidera visualizzare le informazioni, è possibile specificarli quando si esegue il comando **amqsrua** .

Per impostazione predefinita, l'applicazione amqsrua ricerca le statistiche pubblicate dal gestore code nella albero degli argomenti \$SYS/MQ/INFO/QMGR. Altri componenti o applicazioni possono utilizzare un meccanismo simile per la pubblicazione in un punto di partenza dell'argomento differente. Da IBM MQ 9.1.0, è possibile utilizzare il parametro **-p** per specificare dove amqsrua cerca le statistiche per questi altri componenti su Linux e Windows.

-m

È il nome del gestore code. Il gestore code deve essere in esecuzione. Se non si specifica un nome gestore code, vengono visualizzate le statistiche per il gestore code predefinito.

-c

Nome classe. Le pubblicazioni sull'utilizzo delle risorse IBM MQ sono associate a una classe. Le classi rappresentano il livello superiore nella struttura ad albero dei metadati che descrivono le informazioni sull'utilizzo delle risorse disponibili.

CPU

Restituisce informazioni sull'utilizzo della CPU.

DISK

Restituisce informazioni sull'utilizzo del disco.

STATMQI

Restituisce informazioni sull'utilizzo di MQI.

STATQ

Restituisce informazioni sull'utilizzo MQI per coda.

STATAPP

Restituisce informazioni sulle statistiche di utilizzo per l'applicazione specificata.

Consultare “Sviluppo dell'applicazione di monitoraggio delle risorse proprie” a pagina 300 per informazioni su come specificare la struttura ad albero degli argomenti per i metadati e Utilizzo del nome applicazione nei linguaggi di programmazione supportati per i caratteri che è possibile utilizzare nei nomi delle applicazioni.

Nota: Utilizzare l'opzione **-o** per specificare il nome dell'applicazione monitorata per STATAPP e il nome della coda per STATQ.

-t

Nome TYPE. Le pubblicazioni sull'utilizzo delle risorse IBM MQ sono associate a un tipo all'interno di una classe. Ogni pubblicazione include la classe e tipo che consente di trovare le definizioni `class/type/element` e di gestire le pubblicazioni risultanti. Le descrizioni `class/type/element` vengono pubblicate come metadati all'avvio del gestore code.

-o

Il nome dell'oggetto. Le pubblicazioni sull'utilizzo delle risorse sono messaggi PCF costituiti da una sequenza di elementi PCF. Gli elementi PCF pubblicati per ogni coppia classe / tipo vengono pubblicizzati nei metadati. Una foglia viene memorizzata nella struttura ad albero che descrive ciascun elemento, consentendo così l'elaborazione degli elementi.

-p

Prefisso metadati. Specificare un punto di partenza della struttura ad albero degli argomenti in cui **amqsrua** può cercare le statistiche pubblicate dai gestori code. La struttura ad albero degli argomenti predefinita è `$SYS/MQ/INFO/QMGR` ma altri componenti o applicazioni potrebbero pubblicare le statistiche in un punto di partenza differente della struttura ad albero degli argomenti.

-n

Conteggio pubblicazioni. È possibile specificare il numero di prospetti restituiti prima della fine del comando. I dati vengono pubblicati approssimativamente ogni dieci secondi, quindi se si immette un valore di 50, il comando restituisce 50 report oltre 500 secondi. Se non si specifica questo parametro, il comando viene eseguito fino a quando non si verifica un errore o fino a quando il gestore code non viene arrestato.

-s

Coda modello. Facoltativamente, specificare una coda modello da utilizzare. Per impostazione predefinita, **amqsrua** utilizza `SYSTEM.DEFAULT.MODEL.QUEUE` .)

-h

Utilizzo

Procedura

1. Dalla directory `samples`, immettere il seguente comando per visualizzare i dati disponibili per il gestore code:

-  Su Linux, `MQ_INSTALLATION_PATH/samp/bin`:

```
./amqsrua -m QMgrName
```

-  Su Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`:

```
amqsrua -m QMgrName
```

dove *QMgrName* specifica il nome del gestore code che si desidera interrogare. Il gestore code deve essere in esecuzione. Se non si specifica un nome gestore code, viene utilizzato il gestore code predefinito.

Sono disponibili le seguenti opzioni:

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. Dall'elenco delle opzioni CLASS, immettere 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. Dall'elenco delle opzioni TYPE, immettere 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
```

Risultati

L'applicazione di esempio **amqsrua** è stata utilizzata in modo interattivo per esaminare statistiche pubblicate dai gestori code negli argomenti di sistema sotto il prefisso dei metadati \$SYS/MQ/INFO/QMGR.

Nota: Le classi e i tipi di pubblicazioni delle risorse disponibili per i gestori code potrebbero essere diversi a seconda della configurazione, della versione e della piattaforma. Utilizzare **amqsrua** in modo interattivo per trovare le classi, i tipi e gli elementi disponibili per il gestore code specifico.

Operazioni successive

Per sviluppare la propria applicazione di controllo, consultare [“Sviluppo dell'applicazione di monitoraggio delle risorse proprie”](#) a pagina 300.

Attività correlate

[Preparazione ed esecuzione di programmi di esempio su Windows](#)

[Preparazione ed esecuzione di esempi di programmi su UNIX e Linux](#)

Sviluppo dell'applicazione di monitoraggio delle risorse proprie

È possibile sviluppare la propria applicazione per monitorare le risorse di sistema.

Ogni gestore code pubblica i dati di utilizzo delle risorse negli argomenti. Questi dati vengono utilizzati dai sottoscrittori di questi argomenti. Quando un gestore code viene avviato, il gestore code pubblica una serie di messaggi sui meta - argomenti. Questi messaggi descrivono gli argomenti di utilizzo delle risorse supportati dal gestore code e il contenuto dei messaggi pubblicati in tali argomenti. Gli strumenti di gestione possono sottoscrivere i metadati per rilevare quali informazioni sull'utilizzo delle risorse sono disponibili e su quali argomenti, quindi sottoscrivere gli argomenti pubblicizzati.

La struttura ad albero degli argomenti per i metadati ha la seguente struttura:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

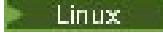


Utilizzare la seguente struttura per i metadati della struttura ad albero degli argomenti:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

per chiarire che si sta specificando la risorsa monitorata, che ha un massimo di 28 caratteri per il nome dell'applicazione, e per evitare confusione con gli attributi del tipo di metadati interni.

Per un elenco delle classi possibili, consultare [“Monitoraggio dell'utilizzo delle risorse del sistema utilizzando il comando amqsrua”](#) a pagina 297.

Il codice sorgente per l'applicazione amqsrua viene fornito come esempio IBM MQ . È possibile utilizzare questa applicazione di esempio come guida per la creazione della propria applicazione di controllo. È possibile recuperare l'origine per l'esempio da un'installazione client IBM MQ . Il file di origine è denominato amqsrua.c e si trova nella directory degli esempi:

-   Su piattaforme AIX and Linux , `MQ_INSTALLATION_PATH/samp/`
-  Su piattaforme Windows , `MQ_INSTALLATION_PATH\tools\c\Samples\`

L'applicazione amqsrua sottoscrive gli argomenti di utilizzo delle risorse IBM MQ e formatta i dati PCF pubblicati risultanti. L'origine dell'applicazione fornisce un esempio di base di come sottoscrivere e utilizzare questo tipo di dati di gestione. L'applicazione amqsrua completa le seguenti attività:

- Crea una sottoscrizione non durevole agli argomenti identificati dai parametri di input.
- Richiama MQGET ripetutamente per richiamare i messaggi dagli argomenti e scrive in stdout.
- Scrive un messaggio per ciascun motivo MQI (diverso da MQRC_NONE).
- Si arresta se è presente un codice di completamento MQI di MQCC_FAILED o quando è stato utilizzato il numero richiesto di pubblicazioni sull'utilizzo delle risorse.

Metriche pubblicate sugli argomenti di sistema

Le metriche sono categorizzate in classi e sottocategorizzate in tipi. Esistono diverse metriche pubblicate in ciascuna classe e tipo di metrica.

Indice analitico

- [“CPU \(unità di elaborazione centrale della piattaforma\)”](#) a pagina 301
- [“DISK \(archivi dati persistenti della piattaforma\)”](#) a pagina 302
- [“STATMQI \(statistiche di utilizzo API\)”](#) a pagina 303
- [“STATQ \(statistiche di utilizzo API per coda\)”](#) a pagina 305
- [“STATAPP \(statistiche di utilizzo per applicazione\)”](#) a pagina 306
- [“NHAREPLICA \(statistiche HA nativa per istanza\)”](#) a pagina 306

Consultare “Monitoraggio dell'utilizzo delle risorse del sistema utilizzando il comando `amqsrva`” a pagina 297 per informazioni su come raccogliere i dati per le opzioni elencate, ad eccezione di NHAREPLICA.

È possibile utilizzare anche il comando `ALTER QMGR` per monitorare `STATMQI` e `STATQ` a livello di gestore code o l'attributo della coda locale `STATQ` per monitorare le singole code; consultare `ALTER QUEUES` per questa opzione.

CPU (unità di elaborazione centrale della piattaforma)

Introduzione

Dove le statistiche fanno riferimento all' *intervallo corrente*, questo è l'intervallo definito nel messaggio pubblicato dal parametro `MQIAMO64_MONITOR_INTERVAL`.

Le statistiche vengono generalmente pubblicate ogni 10 secondi, l' *intervallo pubblicato*, purché esista almeno un sottoscrittore attivo, ma l'intervallo preciso deve essere sempre preso dal messaggio.

Importante: Se non diversamente specificato, le metriche sono valori assoluti per il punto temporale quando vengono acquisite.

SystemSummary (Prestazioni CPU - per piattaforma)

Percentuale tempo CPU utente X%

La percentuale media di tempo (presa nell'ultimo intervallo di 10 secondi) utilizzata dalla CPU quando era in codice non privilegiato.

Percentuale tempo CPU sistema X%

La percentuale media di tempo (utilizzata nell'ultimo intervallo di 10 secondi) utilizzata dalla CPU quando era in codice privilegiato.

Carico CPU - media di un minuto X

La media di carico di un minuto. "Carica media" è un termine a livello di settore, ma il valore esatto riportato potrebbe differire da una piattaforma all'altra.

Carico CPU - media di cinque minuti X

La media di cinque minuti di carico. "Carica media" è un termine a livello di settore, ma il valore esatto riportato potrebbe differire da una piattaforma all'altra.

Carico CPU - media quindici minuti X

Il carico medio di quindici minuti. "Carica media" è un termine a livello di settore, ma il valore esatto riportato potrebbe differire da una piattaforma all'altra.

Riepilogo sistema CPU

Percentuale di RAM libera X%

Byte totali RAM XMB

Percentuale di RAM libera X%

Byte totali RAM XMB

QMGrSummary (Prestazioni CPU - gestore code in esecuzione)

Tempo CPU utente - stima percentuale per il gestore code X%

La percentuale media di tempo (utilizzata nell'ultimo intervallo di 10 secondi) utilizzata dalla CPU quando i processi di questo gestore code erano in codice senza privilegi.

Tempo CPU di sistema - stima percentuale per il gestore code X%

La percentuale media di tempo (utilizzata nell'ultimo intervallo di 10 secondi) utilizzata dalla CPU quando i processi di questo gestore code erano in codice privilegiato.

Byte totali RAM - stima per gestore code XMB

Questa è un'approssimazione della memoria utilizzata dal gestore code.

DISK (archivi dati persistenti della piattaforma)

Il *SystemSummary* e *QMgrSummary* sono valori assoluti al punto di cattura. Consultare [Introduzione](#) per i dettagli dell' *intervallo pubblicato*.

SystemSummary (utilizzo disco - su tutta la piattaforma)

File system errori MQ - byte in uso XMB

MQ errori file system - spazio libero X%

MQ Conteggio file FDC X

File system di traccia MQ - byte in uso XMB

Spazio libero del file system di traccia MQ X%

QMgrSummary (utilizzo disco - gestori code in esecuzione)

File system del gestore code - byte in uso XMB

File system del Gestore code - spazio libero X%

Log (utilizzo disco - log di ripristino gestore code)

Log - byte in uso X

Log - byte max X

Il numero massimo di byte che è possibile scrivere nel log se tutte le estensioni primaria e secondaria erano piene. È inferiore alla dimensione del file system di log

File system di log - byte in uso X

File system di log - byte max X

Byte fisici di log scritti per l'intervallo corrente X.

Consultare [Introduzione](#) per la definizione di *intervallo corrente*.

Byte logici di log scritti per l' *intervallo corrente* X

Latenza scrittura log X uSec

Una media mobile che rappresenta il tempo impiegato da una singola scrittura su disco.

Dove **LogWriteIntegrity**=*TripleWrite*, il numero fisico di byte scritti sul disco è maggiore dei byte logici scritti.

Dimensione di scrittura log X, anch'essa media mobile.

Log occupato dalle estensioni in attesa di archiviazione X.

Pubblicato solo quando **logtype**= *linear* e **LogManagement** = *archive*. Per ulteriori informazioni, consultare [Stanza di log del file qm.ini](#) .

Spazio di log in MB richiesto per il ripristino del supporto X.

Pubblicato solo quando **logtype**= *lineare*.

Spazio di log in MB occupato dalle estensioni riutilizzabili X

Pubblicato solo quando **logtype**= *lineare* e **LogManagement** = *automatico*. Per ulteriori informazioni, consultare [Stanza di log del file qm.ini](#) .

Log - spazio primario corrente in uso X%.

Lo spazio file di log in uso come percentuale dei log primari. Questo valore può essere superiore al 100%.

Log - utilizzo spazio primario del carico di lavoro X%.

La percentuale di spazio file di log in uso come media mobile nella cronologia recente.

STATMQI (statistiche di utilizzo API)

Tutte le statistiche di utilizzo API riflettono le ricorrenze e / o le percentuali per l'intervallo pubblicato. Consultare [Introduzione](#) per la definizione di *intervallo pubblicato*.

Le statistiche riportano un conteggio del numero di chiamate MQI non riuscite, ma non tutte le chiamate MQI non riuscite appaiono in queste statistiche - infatti, le statistiche non vengono registrate per gli errori di tutte le chiamate MQI. Questo perché molti motivi per cui le chiamate MQI non riescono vengono diagnosticati prima che la chiamata MQI raggiunga i valori interni del gestore code in cui vengono registrate le statistiche.

Un esempio di ciò è MQRC_HCONN_ERROR restituito a un'applicazione client. Se un'applicazione client passa un **hconn** non corretto, il client MQ diagnostica l'errore e restituisce MQRC_HCONN_ERROR senza passare la chiamata MQI al gestore code. Pertanto, la chiamata MQI non riuscita non appare mai nelle statistiche registrate dal gestore code.

Le statistiche delle chiamate MQI non riuscite sono utili perché consentono ai clienti di risolvere i problemi delle applicazioni scritte in modo non corretto che generano chiamate MQI non riuscite non necessarie, influenzando le prestazioni. Alcuni esempi di motivi di errore per varie chiamate MQI registrate nelle statistiche:

- MQCONN/MQCONN/MQOPEN restituisce 2035 MQRC_NOT_AUTHORIZED quando viene diagnosticato dal gestore code, non dal client. Ad esempio, l'esecuzione di **amqsput** come nessuno.
- MQPUT/MQPUT1 restituisce 2053 MQRC_Q_FULL perché MAXDEPTH è stato superato.
- MQGET restituisce 2033 MQRC_NO_MSG_AVAILABLE durante l'esplorazione o l'acquisizione distruttiva da una coda vuota
- MQSUBRQ restituisce 2437 MQRC_NO_RETAINED_MSG perché non è presente alcun messaggio conservato

CONNDISC (MQCONN e MQDISC)

Conteggio MQCONN/MQCONN X

Conteggio MQCONN/MQCONN non riuscito X

Connessioni simultanee - contrassegno di limite massimo X

Conteggio MQDISC X

OPENCLOSE (MQOPEN e MQCLOSE)

Conteggio MQOPEN X Y/ sec

Conteggio MQOPEN non riusciti X

Conteggio MQCLOSE X Y/ sec

Conteggio MQCLOSE non riusciti X

INQSET (MQINQ e MQSET)

Conteggio MQINQ X

Conteggio MQINQ non riusciti X

Conteggio MQSET X

Conteggio MQSET non riusciti X

PUT (MQPUT)

Conteggio totale intervallo MQPUT/MQPUT1 X

Numero totale di byte MQPUT/MQPUT1 X Y/ sec

Conteggio MQPUT messaggio non persistente X

Conteggio MQPUT messaggi persistenti X

Conteggio MQPUT non riusciti X

Messaggio non persistente MQPUT1 conteggio X

Conteggio messaggi persistenti MQPUT1 X

Conteggio MQPUT1 non riuscito X

Messaggi non persistenti di inserimento - conteggio byte *X Y*/ sec

Messaggi permanenti di inserimento - conteggio byte *X*

Conteggio MQSTAT *X*

GET (MQGET)

Conteggio totale ricezioni distruttive dell'intervallo *X*

Conteggio totale byte di ricezione distruttivi dell'intervallo *X Y*/ sec

Ricezione distruttiva dei messaggi non persistenti - conteggio *X*

Ricezione distruttiva del messaggio persistente - conteggio *X*

MQGET non riuscito - conteggio *X*

Messaggi non persistenti ricevuti - conteggio byte *X Y*/ sec

Messaggi persistenti ricevuti - conteggio byte *X*

Esplorazione messaggi non persistenti - conteggio *X*

Esplorazione messaggi persistenti - conteggio *X*

Conteggio ricerche non riuscite *X*

Ricerca messaggi non persistenti - conteggio byte *X Y*/ sec

Ricerca messaggi persistenti - conteggio byte *X*

Conteggio messaggi scaduti *X*

Conteggio code eliminate *X*

Conteggio MQCB *X*

Conteggio MQCB non riusciti *X*

Conteggio MQCTL *X*

SYNCPOINT (commit e rollback)

Conteggio commit *X*

Conteggio rollback *X*

SUBSCRIBE (sottoscrizione)

Crea conteggio sottoscrizioni durevoli *X*

Modifica conteggio sottoscrizioni durevoli *X*

Ripristina conteggio sottoscrizioni durevoli *X*

Crea conteggio sottoscrizioni non durevoli *X*

Conteggio sottoscrizioni create / alter/resume *X* non riuscito

Elimina conteggio sottoscrizioni durevoli *X*

Elimina conteggio sottoscrizioni non durevoli *X*

Conteggio errori eliminazione sottoscrizione *X*

Conteggio MQSUBRQ: *X*

Conteggio MQSUBRQ non riusciti *X*

Sottoscrittore durevole - limite massimo *X*

Sottoscrittore durevole - limite minimo *X*

Sottoscrittore non durevole - high water mark *X*

Sottoscrittore non durevole - low water mark *X*

PUBLISH (pubblicazione)

Argomento MQPUT/MQPUT1 intervallo totale *X*

Byte argomento totali di intervallo immessi *X Y*/ sec

Pubblicato per i sottoscrittori - conteggio messaggi *X*

Pubblicato per sottoscrittori - conteggio byte *X*

Argomento non persistente MQPUT/MQPUT1 conteggio X

Persistente - argomento MQPUT/MQPUT1 conteggio X

Argomento non riuscito MQPUT/MQPUT1 conteggio X

STATQ (statistiche di utilizzo API per coda)

GENERALE (Generale)

messaggi scaduti X (spostati da GET per IBM MQ 9.3.0 e versioni successive CD)

conteggio code eliminate X (spostate da GET per IBM MQ 9.3.0 e versioni successive CD)

tempo medio della coda X uSec (spostato da GET per IBM MQ 9.3.0 e versioni successive CD)

Profondità coda X (spostata da GET per IBM MQ 9.3.0 e versioni successive CD)

OPENCLOSE (MQOPEN e MQCLOSE)

Conteggio MQOPEN X

Conteggio MQCLOSE X

INQSET (MQINQ e MQSET)

Conteggio MQINQ X

Conteggio MQSET X

PUT (MQPUT e MQPUT1)

MQPUT/MQPUT1 conteggio X

Conteggio byte MQPUT X

Numero di messaggi non persistenti MQPUT X

Conteggio messaggi persistenti MQPUT X

conteggio MQPUT di cui è stato eseguito il rollback X

MQPUT1 conteggio messaggi non persistenti X

MQPUT1 conteggio messaggi persistenti X

conteggio byte non persistenti X

conteggio byte persistenti X

contesa di blocco X%

La percentuale di tentativi di blocco della coda che hanno determinato l'attesa di un altro processo per rilasciare prima il blocco. La diminuzione del conflitto di blocco probabilmente aumenterà la velocità di trasmissione massima del sistema poiché l'utilizzo di un blocco che non è attualmente bloccato è più efficiente rispetto all'attesa del rilascio di un blocco.

coda evitata inserisce X%

Se un messaggio viene inserito in una coda quando è presente un getter in attesa, potrebbe non essere necessario accodare il messaggio poiché è possibile che venga trasmesso immediatamente al getter. Quindi si dice che questo messaggio abbia evitato la coda, e "inserimenti evitati dalla coda" è il conteggio di questi messaggi. È probabile che l'aumento della riduzione della coda aumenti la velocità di trasmissione massima del sistema, poiché evita il costo di inserire il messaggio nella coda e di disattivarlo di nuovo.

byte evitati dalla coda X%

Se un messaggio viene inserito in una coda quando è presente un getter in attesa, potrebbe non essere necessario accodare il messaggio poiché è possibile che venga trasmesso immediatamente al getter. Si dice che questo messaggio abbia evitato la coda e che "byte evitati dalla coda" sia il conteggio di tali byte. È probabile che l'aumento della riduzione della coda aumenti la velocità di trasmissione massima del sistema, poiché evita il costo di inserire il messaggio nella coda e di disattivarlo di nuovo.

GET (MQGET)

Conteggio MQGET X

Conteggio byte MQGET X
 conteggio messaggi non persistenti MQGET distruttivi X
 conteggio messaggi permanenti MQGET distruttivi X
 conteggio MQGET di cui è stato eseguito il rollback X
 conteggio byte non persistenti MQGET distruttivi X
 conteggio byte permanenti MQGET distruttivi X
 Conteggio messaggi non persistenti esplorazione MQGET X
 Conteggio messaggi persistenti di esplorazione MQGET X
 Conteggio byte non persistenti esplorazione MQGET X
 Conteggio byte persistenti MQGET browse X
 messaggi scaduti X (spostati in GENERAL da IBM MQ 9.3)
 conteggio code eliminate X (spostate in GENERAL da IBM MQ 9.3)
 tempo medio della coda X uSec (spostato in GENERAL da IBM MQ 9.3)
 Profondità coda X (spostata in GENERAL da IBM MQ 9.3)
 errore MQGET distruttivo X
 MQGET distruttivo non riesce con MQRC_NO_MSG_AVAILABLE X
 MQGET distruttivo non riesce con MQRC_TRUNCATED_MSG_FAILED X
 Ricerca MQGET non riuscita X
 Ricerca MQGET non riuscita con MQRC_NO_MSG_AVAILABLE X
 L'esplorazione MQGET ha esito negativo con MQRC_TRUNCATED_MSG_FAILED X

STATAPP (statistiche di utilizzo per applicazione)

INSTANCE (statistiche istanza)

Conteggio istanze X assoluto
 Conteggio istanze mobili X assoluto
 Conteggio ammanco istanza X assoluto
 Istanze avviate X intervallo
 L'istanza in uscita avviata sposta X intervallo
 Spostamenti di istanze in uscita completati X intervallo
 Istanze terminate durante la riconnessione X intervallo
 Istanze terminate X intervallo

NHAREPLICA (statistiche HA nativa per istanza)

REPLICATION (statistiche di replica)

Byte di log sincroni inviati X
 Byte di log catch - up inviati X
 ▶ V 9.4.0 Byte di log compressi sincroni inviati X
 ▶ V 9.4.0 Byte di log compressi di recupero inviati X
 ▶ V 9.4.0 Byte di log non compressi sincroni inviati X
 ▶ V 9.4.0 Byte di log non compressi catch - up inviati X
 Latenza media riconoscimento scrittura log X uSec
 Dimensione media riconoscimento scrittura log X
 Byte backlog X

Informazioni correlate

[Metriche pubblicate quando si utilizza IBM MQ Operator](#)

Monitoraggio in tempo reale

Il monitoraggio in tempo reale è una tecnica che consente di determinare lo stato corrente di code e canali all'interno di un gestore code. Le informazioni restituite sono accurate nel momento in cui è stato immesso il comando.

Sono disponibili diversi comandi che, quando vengono emessi, restituiscono informazioni in tempo reale su code e canali. Le informazioni possono essere restituite per una o più code o canali e possono variare in quantità. Il monitoraggio in tempo reale può essere utilizzato nelle attività seguenti:

- Aiutare gli amministratori di sistema a comprendere lo stato stazionario del proprio sistema IBM MQ . Ciò consente di diagnosticare il problema se si verifica un problema nel sistema.
- Determinazione della condizione del proprio gestore code in qualsiasi momento, anche se non è stato rilevato alcun evento o problema specifico.
- Assistenza nella determinazione della causa di un problema nel sistema.

Con il monitoraggio in tempo reale, le informazioni possono essere restituite per code o canali. La quantità di informazioni in tempo reale restituite è controllata dagli attributi gestore code, coda e canale.

- È possibile monitorare una coda immettendo dei comandi per assicurarsi che la coda venga gestita correttamente. Prima di poter utilizzare alcuni attributi della coda, è necessario abilitarli per il controllo in tempo reale.
- Si monitora un canale immettendo comandi per assicurarsi che il canale sia in esecuzione correttamente. Prima di utilizzare alcuni degli attributi del canale, è necessario abilitarli per il monitoraggio in tempo reale.

Il monitoraggio in tempo reale di code e canali è in aggiunta e separato dal monitoraggio di eventi di canale e prestazioni.

Attributi che controllano il monitoraggio in tempo reale

Alcuni attributi di stato della coda e del canale contengono le informazioni di monitoraggio, se il monitoraggio in tempo reale è abilitato. Se il monitoraggio in tempo reale non è abilitato, non vengono conservate informazioni di monitoraggio in questi attributi di monitoraggio. Gli esempi dimostrano come utilizzare questi attributi di stato della coda e del canale.

È possibile abilitare o disabilitare il controllo in tempo reale per singole code o canali o per più code o canali. Per controllare le singole code o canali, impostare l'attributo della coda MONQ o l'attributo del canale MONCHL, per abilitare o disabilitare il controllo in tempo reale. Per controllare più code o canali insieme, abilitare o disabilitare il monitoraggio in tempo reale a livello di gestore code utilizzando gli attributi gestore code MONQ e MONCHL. Per tutti gli oggetti coda e canale con un attributo di controllo specificato con il valore predefinito, QMGR, il monitoraggio in tempo reale è controllato a livello di gestore code.

I canali mittente del cluster definiti automaticamente non sono oggetti IBM MQ , quindi non hanno attributi allo stesso modo degli oggetti del canale. Per controllare i canali mittenti del cluster definiti automaticamente, utilizzare l'attributo del gestore code MONACLS. Questo attributo determina se i canali mittenti del cluster definiti automaticamente all'interno di un gestore code sono abilitati o disabilitati per il monitoraggio del canale.

Per il monitoraggio in tempo reale dei canali, è possibile impostare l'attributo MONCHL su uno dei tre livelli di monitoraggio: basso, medio o alto. È possibile impostare il livello di monitoraggio a livello di oggetto o a livello di gestore code. La scelta del livello dipende dal sistema. La raccolta dei dati di monitoraggio potrebbe richiedere alcune istruzioni relativamente costose dal punto di vista computazionale, come ad esempio l'ottenimento dell'ora del sistema. Per ridurre l'effetto del monitoraggio in tempo reale, le opzioni di monitoraggio medio e basso misurano un campione di dati

a intervalli regolari piuttosto che raccogliere dati per tutto il tempo. Tabella 30 a pagina 308 riepiloga i livelli di controllo disponibili per il controllo in tempo reale dei canali:

<i>Tabella 30. Livelli di controllo</i>		
Livello	Descrizione	Utilizzo
Bassa	Misurare un piccolo campione di dati, ad intervalli regolari.	Per oggetti che elaborano un volume elevato di messaggi.
Medio	Misurare un campione di dati, ad intervalli regolari.	Per la maggior parte degli oggetti.
Elevato	Misurare tutti i dati, a intervalli regolari.	Per gli oggetti che elaborano solo pochi messaggi al secondo, su cui sono importanti le informazioni più recenti.

Per il monitoraggio in tempo reale delle code, è possibile impostare l'attributo MONQ su uno dei tre livelli di monitoraggio, basso, medio o alto. Tuttavia, non esiste alcuna distinzione tra questi valori. Tutti i valori abilitano la raccolta dati, ma non influiscono sulla dimensione del campione.

Esempi

I seguenti esempi illustrano come impostare gli attributi di coda, canale e gestore code necessari per controllare il livello di controllo. Per tutti gli esempi, quando il monitoraggio è abilitato, gli oggetti coda e canale hanno un medio livello di monitoraggio.

1. Per abilitare il monitoraggio di code e canali per tutte le code e canali a livello di gestore code, utilizzare i seguenti comandi:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

2. Per abilitare il monitoraggio per tutte le code e canali, ad eccezione della coda locale, Q1e del canale mittente, QM1 . TO . QM2, utilizzare i seguenti comandi:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

3. Per disabilitare il monitoraggio di code e canali per tutte le code e canali, ad eccezione della coda locale, Q1e del canale mittente, QM1 . TO . QM2, utilizzare i comandi riportati di seguito:

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

4. Per disabilitare il monitoraggio di code e canali per tutte le code e canali, indipendentemente dai singoli attributi oggetto, utilizzare il seguente comando:

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

5. Per controllare le funzioni di controllo dei canali mittenti del cluster definiti automaticamente, utilizzare i seguenti comandi:

```
ALTER QMGR MONACLS(MEDIUM)
```

6. Per specificare che i canali mittente del cluster definiti automaticamente devono utilizzare l'impostazione del gestore code per il monitoraggio del canale, utilizzare il seguente comando:

Concetti correlati

[“Monitoraggio in tempo reale” a pagina 307](#)

Il monitoraggio in tempo reale è una tecnica che consente di determinare lo stato corrente di code e canali all'interno di un gestore code. Le informazioni restituite sono accurate nel momento in cui è stato immesso il comando.

[“Using IBM MQ online monitoring” a pagina 328](#)

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

Attività correlate

[“Visualizzazione dei dati di monitoraggio di code e canali” a pagina 309](#)

Per visualizzare le informazioni di monitoraggio in tempo reale per una coda o un canale, utilizzare il comando IBM MQ Explorer o MQSC appropriato. Alcuni campi di monitoraggio visualizzano una coppia di valori indicatore separati da virgole, che consentono di monitorare il funzionamento del gestore code. Gli esempi dimostrano come visualizzare i dati di monitoraggio.

[Visualizzazione e modifica degli attributi del gestore code](#)

[Monitoraggio \(MONCHL\)](#)

Visualizzazione dei dati di monitoraggio di code e canali

Per visualizzare le informazioni di monitoraggio in tempo reale per una coda o un canale, utilizzare il comando IBM MQ Explorer o MQSC appropriato. Alcuni campi di monitoraggio visualizzano una coppia di valori indicatore separati da virgole, che consentono di monitorare il funzionamento del gestore code. Gli esempi dimostrano come visualizzare i dati di monitoraggio.

Informazioni su questa attività

I campi di monitoraggio che visualizzano una coppia di valori separati da una virgola forniscono indicatori a breve e a lungo termine per il tempo misurato da quando è stato abilitato il monitoraggio per l'oggetto o da quando è stato avviato il gestore code:

- L'indicatore a breve termine è il primo valore della coppia ed è calcolato in maniera tale che le misurazioni più recenti hanno una ponderazione più elevata e avranno un effetto maggiore su questo valore. Ciò fornisce un'indicazione dell'andamento recente delle misurazioni effettuate.
- L'indicatore a lungo termine nel secondo valore della coppia ed è calcolato in modo tale che le misurazioni più recenti non ricevano una ponderazione così elevata. Ciò fornisce un'indicazione dell'attività a più lungo termine sulle prestazioni di una risorsa.

Questi valori di indicatore sono più utili per rilevare le modifiche nel funzionamento del gestore code. Ciò richiede la conoscenza dei tempi che questi indicatori mostrano quando sono in uso normale, al fine di rilevare aumenti in questi tempi. Raccogliendo e controllando regolarmente questi valori è possibile rilevare le fluttuazioni nel funzionamento del gestore code. Ciò può indicare una modifica delle prestazioni.

Ottenere le informazioni di monitoraggio in tempo reale come segue:

Procedura

1. Per visualizzare le informazioni di controllo in tempo reale per una coda, utilizzare il comando IBM MQ Explorer o MQSC `DISPLAY QSTATUS`, specificando il parametro facoltativo `MONITOR`.
2. Per visualizzare le informazioni di monitoraggio in tempo reale per un canale, utilizzare il comando IBM MQ Explorer o il comando MQSC `DISPLAY CHSTATUS`, specificando il parametro facoltativo `MONITOR`.

Esempio

La coda, Q1, ha l'attributo MONQ impostato sul valore predefinito, QMGR, e il gestore code proprietario della coda ha l'attributo MONQ impostato su MEDIUM. Per visualizzare i campi di controllo raccolti per questa coda, utilizzare il comando seguente:

```
DISPLAY QSTATUS(Q1) MONITOR
```

I campi di monitoraggio e il livello di controllo della coda, Q1 , vengono visualizzati come segue:

```
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)
```

Il canale mittente, QM1 . TO . QM2, ha l'attributo MONCHL impostato sul valore predefinito, QMGR, e il gestore code proprietario della coda ha l'attributo MONCHL impostato su MEDIUM. Per visualizzare i campi di monitoraggio raccolti per questo canale mittente, utilizzare il seguente comando:

```
DISPLAY CHSTATUS(QM1.TO.QM2) MONITOR
```

I campi di controllo e il livello di controllo del canale mittente, QM1 . TO . QM2 vengono visualizzati come segue:

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

Concetti correlati

[“Monitoraggio in tempo reale” a pagina 307](#)

Il monitoraggio in tempo reale è una tecnica che consente di determinare lo stato corrente di code e canali all'interno di un gestore code. Le informazioni restituite sono accurate nel momento in cui è stato immesso il comando.

Riferimenti correlati

[STATOQVISUALIZZAZIONE](#)

Code di monitoraggio

Utilizzare questa pagina per visualizzare le attività che consentono di risolvere un problema con una coda e l'applicazione che esegue tale coda. Sono disponibili varie opzioni di monitoraggio per determinare il problema

Di frequente, il primo segno di un problema con una coda che viene gestita è che il numero di messaggi sulla coda (CURDEPTH) aumenta. Se si prevede un aumento in determinate ore del giorno o in determinati workload, un numero crescente di messaggi potrebbe non indicare un problema. Tuttavia, se non si dispone di una spiegazione per il numero crescente di messaggi, è possibile indagare sulla causa.

È possibile che si disponga di una coda dell'applicazione in cui si è verificato un problema con l'applicazione o di una coda di trasmissione in cui si è verificato un problema con il canale. Le opzioni di monitoraggio aggiuntive sono disponibili quando l'applicazione che serve la coda è un canale.

I seguenti esempi analizzano i problemi con una particolare coda, denominata Q1, e descrivono i campi visualizzati nell'emissione di vari comandi:

Determinazione se l'applicazione ha la coda aperta

In caso di problemi con una coda, verificare se l'applicazione ha la coda aperta

Informazioni su questa attività

Effettuare le seguenti operazioni per stabilire se l'applicazione ha la coda aperta:

Procedura

1. Verificare che l'applicazione in esecuzione rispetto alla coda sia l'applicazione prevista. Immettere il seguente comando per la coda in questione:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

Nell'output, esaminare il campo APPLTAG e verificare che sia visualizzato il nome della propria applicazione. Se il nome dell'applicazione non viene visualizzato o se non è presente alcun output, avviare l'applicazione.

2. Se la coda è una coda di trasmissione, esaminare l'emissione nel campo CHANNEL.
Se il nome del canale non viene visualizzato nel campo CHANNEL, determinare se il canale è in esecuzione.
3. Verificare che l'applicazione in esecuzione sulla coda abbia la coda aperta per l'input. Emetti il seguente comando:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Nell'emissione, esaminare il campo IPPROCS per vedere se un'applicazione ha la coda aperta per l'immissione. Se il valore è 0 e questa è una coda dell'applicazione utente, assicurarsi che l'applicazione apra la coda per l'input per richiamare i messaggi dalla coda.

Verifica che i messaggi sulla coda siano disponibili

Se sulla coda è presente un numero elevato di messaggi e l'applicazione non sta elaborando tali messaggi, verificare se i messaggi sulla coda sono disponibili per l'applicazione

Informazioni su questa attività

Effettuare le seguenti operazioni per individuare il motivo per cui l'applicazione non sta elaborando i messaggi dalla coda:

Procedura

1. Assicurarsi che l'applicazione non stia richiedendo un ID messaggio specifico o un ID correlazione quando deve elaborare tutti i messaggi sulla coda.
2. Anche se la grandezza corrente della coda potrebbe mostrare che c'è un numero crescente di messaggi nella coda, alcuni messaggi nella coda potrebbero non essere disponibili per essere ricevuti da un'applicazione, poiché non ne è stato eseguito il commit; la grandezza corrente include il numero di MQPUT di messaggi non sottoposti a commit nella coda. Emetti il seguente comando:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Nell'output, esaminare il campo UNCOM per verificare se nella coda sono presenti messaggi senza commit.

3. Se l'applicazione sta tentando di richiamare i messaggi dalla coda, verificare se l'applicazione di inserimento sta eseguendo correttamente il commit dei messaggi. Immettere il seguente comando per individuare i nomi delle applicazioni che stanno inserendo i messaggi in questa coda:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) OPENTYPE(OUTPUT)
```

4. Quindi immettere il seguente comando, inserendo in *appltag* il valore APPLTAG dall'output del precedente comando:

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

Mostra quando è stata avviata l'unità di lavoro e consente di rilevare se l'applicazione sta creando un'unità di lavoro di lunga durata. Se l'applicazione di inserimento è un canale, è possibile esaminare il motivo per cui il completamento di un batch sta impiegando molto tempo.

Verifica se l'applicazione sta richiamando i messaggi dalla coda

In caso di problemi con una coda e con l'applicazione che la utilizza, verificare se l'applicazione sta ricevendo messaggi dalla coda

Informazioni su questa attività

Per controllare se l'applicazione sta richiamando i messaggi dalla coda, eseguire i seguenti controlli:

Procedura

1. Verificare che l'applicazione in esecuzione sulla coda stia effettivamente elaborando i messaggi dalla coda. Emetti il seguente comando:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Nell'output, esaminare i campi LGETDATE e LGETTIME che mostrano quando è stata eseguita l'ultimo richiamo dalla coda.

2. Se l'ultimo richiamo da questa coda è stato più recente del previsto, assicurarsi che l'applicazione stia elaborando correttamente i messaggi.

Se l'applicazione è un canale, verificare se i messaggi si stanno spostando attraverso tale canale

Come determinare se l'applicazione è in grado di elaborare i messaggi abbastanza velocemente

Se i messaggi si stanno creando sulla coda, ma gli altri controlli non hanno rilevato alcun problema di elaborazione, verificare che l'applicazione possa elaborare i messaggi abbastanza velocemente. Se l'applicazione è un canale, verificare che il canale possa elaborare i messaggi abbastanza velocemente.

Informazioni su questa attività

Per determinare se l'applicazione sta elaborando i messaggi abbastanza velocemente, eseguire i seguenti test:

Procedura

1. Immettere periodicamente il seguente comando per raccogliere i dati sulle prestazioni della coda:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```


Se i valori negli indicatori QTIME sono elevati o sono in aumento nel periodo e si è già esclusa la possibilità di eseguire a lungo le unità di lavoro controllando che i messaggi sulla coda siano disponibili, l'applicazione di richiamo potrebbe non stare al passo con le applicazioni di inserimento.

2. Se l'applicazione di richiamo non è in grado di tenere il passo con le applicazioni di inserimento, considerare la possibilità di aggiungere un'altra applicazione di richiamo per elaborare la coda.

Se è possibile aggiungere un'altra applicazione di richiamo dipende dalla progettazione dell'applicazione e se la coda può essere condivisa da più di un'applicazione. Funzioni quali il raggruppamento dei messaggi o il richiamo in base all'ID di correlazione potrebbero aiutare a garantire che due applicazioni possano elaborare una coda contemporaneamente.

Controllo della coda quando la profondità corrente non aumenta

Anche se la profondità corrente della coda non è in aumento, potrebbe essere utile monitorare la coda per controllare se l'applicazione sta elaborando correttamente i messaggi.

Informazioni su questa attività

Per raccogliere i dati sulle prestazioni relativi alla coda: immettere periodicamente il seguente comando:

Procedura

Immettere periodicamente il seguente comando:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

Nell'output, se il valore in MSGAGE aumenta nel periodo di tempo e l'applicazione è progettata per elaborare tutti i messaggi, ciò potrebbe indicare che alcuni messaggi non vengono elaborati affatto.

Monitoraggio dei canali

Utilizzare questa pagina per visualizzare le attività che consentono di risolvere un problema con una coda di trasmissione e il canale che fornisce tale coda. Sono disponibili diverse opzioni di monitoraggio del canale per determinare il problema.

Di frequente, il primo segno di un problema con una coda che viene gestita è che il numero di messaggi sulla coda (CURDEPTH) aumenta. Se si prevede un aumento in determinate ore del giorno o in determinati workload, un numero crescente di messaggi potrebbe non indicare un problema. Tuttavia, se non si dispone di una spiegazione per il numero crescente di messaggi, è possibile indagare sulla causa.

È possibile che si sia verificato un problema con il canale che fornisce servizi a una coda di trasmissione. Sono disponibili diverse opzioni di monitoraggio del canale che consentono di determinare il problema.

I seguenti esempi analizzano problemi con una coda di trasmissione denominata QM2 e un canale denominato QM1.TO.QM2. Questo canale viene utilizzato per inviare messaggi dal gestore code, QM1, al gestore code, QM2. La definizione di canale sul gestore code QM1 è un canale mittente o server e la definizione di canale sul gestore code, QM2, è un canale ricevente o richiedente.

Come determinare se il canale è in esecuzione

In caso di problemi con una coda di trasmissione, verificare se il canale è in esecuzione.

Informazioni su questa attività

Effettuare le seguenti operazioni per controllare lo stato del canale che serve la coda di trasmissione:

Procedura

1. Immettere il seguente comando per individuare quale canale si prevede di elaborare la coda di trasmissione QM2:

```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```

In questo esempio, l'output di questo comando mostra che il canale che serve la coda di trasmissione è QM1.TO.QM2

2. Immettere il seguente comando per determinare lo stato del canale, QM1.TO.QM2:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

3. Esaminare il campo STATUS dell'output dal comando **CHSTATUS** :

- Se il valore del campo STATUS è RUNNING, verificare che il canale stia spostando i messaggi
- Se l'output del comando non mostra alcuno stato o il valore del campo STATUS è STOPPED, RETRY, BINDING o REQUESTING, eseguire il passo appropriato, come riportato di seguito:

4. Opzionale: Se il valore del campo STATUS non mostra alcuno stato, il canale è inattivo, quindi effettuare le seguenti operazioni:

- a) Se il canale deve essere stato avviato automaticamente da un trigger, verificare che i messaggi sulla coda di trasmissione siano disponibili.

Se ci sono messaggi disponibili sulla coda di trasmissione, controllare che le impostazioni del trigger sulla coda di trasmissione siano corrette.

- b) Immettere il seguente comando per avviare nuovamente il canale manualmente:

```
START CHANNEL(QM1.TO.QM2)
```


5. Opzionale: Se il valore del campo STATUS è STOPPED, effettuare le seguenti operazioni:

- a) Controllare i log degli errori per stabilire perché il canale è stato arrestato. Se il canale si è arrestato a causa di un errore, correggere il problema.

Verificare inoltre che il canale disponga di valori specificati per gli attributi dei tentativi: *SHORTRTY* e *LONGRTY*. In caso di errori transitori, ad esempio errori di rete, il canale tenterà di riavviare automaticamente.

- b) Immettere il seguente comando per avviare nuovamente il canale manualmente:

```
START CHANNEL(QM1.TO.QM2)
```

 Su IBM MQ for z/OS, è possibile rilevare quando un utente arresta un canale utilizzando i messaggi di evento comando.

6. Opzionale: Se il valore del campo STATUS è RETRY, effettuare le seguenti operazioni:

- a) Controllare i log degli errori per identificare l'errore, quindi correggere il problema.

- b) Immettere il seguente comando per avviare nuovamente il canale manualmente:

```
START CHANNEL(QM1.TO.QM2)
```

o attendono che il canale si connetta correttamente al successivo tentativo.

7. Opzionale: Se il valore del campo STATUS è BINDING o REQUESTING, il canale non si è ancora connesso correttamente al partner. Effettuare le seguenti operazioni:

- a) Immettere il seguente comando, ad entrambe le estremità del canale, per determinare lo stato secondario del canale:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```


Nota:

i) In alcuni casi potrebbe essere presente uno stato secondario solo ad un'estremità del canale.

ii) Molti sottostati sono transitori, quindi emettere il comando alcune volte per rilevare se un canale è bloccato in un particolare sottostato.

- b) Controllare [Tabella 31 a pagina 315](#) per determinare quale azione eseguire:

Tabella 31. Sottostati visualizzati con bind di stato o richiesta

Inizializzazione stato secondario MCA ¹	Stato secondario MCA di risposta ²	Note
NameServer		L'MCA di inizializzazione è in attesa del completamento di una richiesta del nome server. Verificare che sia stato specificato il nome host corretto nell'attributo del canale, CONNAME, e che i server dei nomi siano impostati correttamente.
SCYEXIT	SCYEXIT	Gli MCA sono attualmente <i>in conversazione</i> tramite un'uscita di sicurezza. Per ulteriori informazioni, consultare “Come determinare se il canale è in grado di elaborare i messaggi abbastanza velocemente” a pagina 317.
	CHADEXIT	L'uscita di definizione automatica del canale è attualmente in esecuzione. Per ulteriori informazioni, consultare “Come determinare se il canale è in grado di elaborare i messaggi abbastanza velocemente” a pagina 317.
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	Le uscite vengono chiamate all'avvio del canale per MQXR_INIT. Esaminare l'elaborazione in questa parte dell'uscita se questa operazione richiede molto tempo. Per ulteriori informazioni, consultare “Come determinare se il canale è in grado di elaborare i messaggi abbastanza velocemente” a pagina 317.
Serializza	Serializza	Questo stato secondario si riferisce solo ai canali con una disposizione SHARED.
COLLEGA		Questo stato secondario viene visualizzato se si verifica un ritardo nella connessione a causa di una configurazione di rete non corretta.
HANDSHAKE	HANDSHAKE	Un handshake TLS consiste in un numero di invii e ricezioni. Se i tempi di rete sono lenti o la connessione ai CRL di ricerca è lenta, ciò influisce sul tempo impiegato per eseguire l'handshake.  Su IBM MQ for z/OS questo stato secondario può anche essere indicativo di non avere un numero sufficiente di SSLTASKS.

Note:

- i) L'MCA iniziale è la fine del canale che ha avviato la conversazione. Possono essere mittenti, mittenti del cluster, server completi e richiedenti. In una coppia server - richiedente, è la fine da cui è stato avviato il canale.
- ii) L'MCA rispondente è la fine del canale che ha risposto alla richiesta di avviare la conversazione. Può trattarsi di ricevitori, ricevitori cluster, richiedenti (quando viene avviato il server o il mittente), server (quando viene avviato il richiedente) e mittenti (in una coppia di canali di call-back richiedente - mittente).

Verifica che il canale stia spostando i messaggi

In caso di problemi con una coda di trasmissione, verificare che il canale stia spostando i messaggi

Prima di iniziare

Immettere il comando DIS CHSTATUS(QM1.T0.QM2) ALL. Se il valore del campo STATUS è RUNNING, il canale si è connesso correttamente al sistema partner.

Verificare che non vi siano messaggi di cui non è stato eseguito il commit nella coda di trasmissione, come descritto in [“Verifica che i messaggi sulla coda siano disponibili”](#) a pagina 311.

Informazioni su questa attività

Se ci sono messaggi disponibili per il canale da richiamare e inviare, effettuare le verifiche riportate di seguito:

Procedura

1. Nell'output del comando di visualizzazione dello stato del canale, DIS CHSTATUS(QM1.TO.QM2) ALL, esaminare i seguenti campi:

MSGS

Il numero di messaggi inviati o ricevuti (o, per i canali di connessione server, il numero di chiamate MQI gestite) durante questa sessione (a partire dall'avvio del canale).

BUFSENT

Numero di buffer di trasmissione inviati. Ciò include le trasmissioni per inviare solo informazioni di controllo.

BYTSENT

Il numero di byte inviati durante questa sessione (dall'avvio del canale). Tale valore include le informazioni di controllo inviate dall'MCA (Message Channel Agent).

LSTMSGDA

La data in cui è stato inviato l'ultimo messaggio o in cui è stata gestita la chiamata MQI, consultare LSTMSGTI.

LSTMSGTI

L'ora in cui è stato inviato l'ultimo messaggio o è stata gestita la chiamata MQI. Per un mittente o un server, questa è l'ora in cui è stato inviato l'ultimo messaggio (l'ultima parte di esso se è stato precedentemente suddiviso). Per un canale richiedente o ricevente, questa è l'ora in cui è stato ricevuto l'ultimo messaggio (o l'ultima sua parte) nella coda di destinazione. Per un canale di connessione server, questa è l'ora in cui è stata completata l'ultima chiamata MQI.

CURMSGGS

Per un canale di invio, questo è il numero di messaggi che sono stati inviati nel batch corrente. Per un canale di ricezione, questo è il numero di messaggi che sono stati ricevuti nel batch corrente. Il valore viene reimpostato su zero, sia per il canale di invio che per quello di ricezione, quando il batch viene completato.

STATO

Lo stato del canale, che può essere Starting, Binding, Initializing, Running, Stopping, Retrying, Paused, Stopped o Requesting.

SUBSTATE

L'azione eseguita correntemente dal canale.

INDOUBT

Indica se il canale è attualmente in stato incerto. Questo è solo YES mentre l'invio di Message Channel Agent è in attesa di un riconoscimento che un batch di messaggi inviati sia stato ricevuto correttamente. Esso è NO in tutti gli altri casi, compreso il periodo in cui i messaggi sono stati inviati ma prima che sia stata richiesta la notifica di ricezione. Per un canale di ricezione il valore è sempre NO.

2. Determinare se il canale ha inviato messaggi dall'avvio. Se ne è stato inviato uno, determinare quando è stato inviato l'ultimo messaggio.
3. Il canale potrebbe aver avviato un batch che non è stato ancora completato, come indicato da un valore diverso da zero in CURMSGGS. Se INDOUBT è YES, il canale è in attesa di ricevere il riconoscimento che l'altra estremità del canale ha ricevuto il batch. Esaminare il campo SUBSTATE nell'output e fare riferimento a [Tabella 32 a pagina 317](#):

Tabella 32. Sottostazioni MCA mittente e ricevente

Stato secondario mittente	SUBSTATE destinatario	Note
MQGET	Ricezione	Stati normali di un canale inattivo.
INVIA	Ricezione	Il SEND è di solito uno stato transitorio. Se viene visualizzato SEND indica che i buffer del protocollo di comunicazione si sono riempiti. Ciò può indicare un problema di rete.
Ricezione		Se il mittente viene visualizzato nello stato secondario RECEIVE per un periodo di tempo qualsiasi, è in attesa di una risposta, a un completamento batch o a un heartbeat. È possibile verificare il motivo per cui un batch impiega molto tempo per essere completato.

Nota: È anche possibile determinare se il canale può elaborare i messaggi abbastanza velocemente, soprattutto se il canale ha uno stato secondario associato all'elaborazione dell'uscita.

Verifica del motivo per cui un batch impiega molto tempo per essere completato

I motivi per cui il completamento di un batch può richiedere molto tempo includono una rete lenta o un canale sta utilizzando l'elaborazione dei tentativi dei messaggi.

Informazioni su questa attività

Quando un canale mittente ha inviato un batch di messaggi, attende la conferma del batch dal destinatario, a meno che il canale non sia pipeline. I fattori descritti in questa attività possono influire sul tempo di attesa del canale mittente.

Procedura

- Verificare se la rete è lenta.

Il valore NETTIME è il tempo, visualizzato in microsecondi, impiegato per inviare una richiesta di fine batch all'estremità remota del canale e ricevere una risposta meno il tempo per elaborare la richiesta di fine batch. Questo valore può essere elevato per uno dei seguenti motivi:

- La rete è lenta. Una rete lenta può influire sul tempo necessario per completare un batch. Le misurazioni che risultano negli indicatori per il campo NETTIME vengono misurate al termine di un batch. Tuttavia, il primo batch interessato da un rallentamento della rete non è indicato con una modifica del valore NETTIME poiché viene misurato alla fine del batch.
- Le richieste vengono accodate all'estremità remota, ad esempio, un canale può ritardare un'operazione di inserimento oppure una richiesta di inserimento potrebbe essere lenta a causa dell'I/O della serie di pagine. Una volta completate le richieste in coda, viene misurata la durata della fine della richiesta batch. Quindi, se si ottiene un valore NETTIME elevato, verificare la presenza di un'elaborazione insolita all'estremità remota.

- Verificare se il canale sta utilizzando un nuovo tentativo di messaggio.

Se il canale ricevente non riesce a inserire un messaggio in una coda di destinazione, potrebbe utilizzare l'elaborazione dei tentativi del messaggio, piuttosto che inserire immediatamente il messaggio in una coda di messaggi non recapitabili. L'elaborazione dei tentativi può causare un rallentamento del batch. Tra i tentativi MQPUT, il canale avrà STATUS (PAUSED), che indica che è in attesa del superamento dell'intervallo di tentativi del messaggio.

Come determinare se il canale è in grado di elaborare i messaggi abbastanza velocemente

Se i messaggi si stanno costruendo sulla coda di trasmissione, ma non sono stati rilevati problemi di elaborazione, determinare se il canale è in grado di elaborare i messaggi abbastanza velocemente.

Prima di iniziare

Immettere il seguente comando ripetutamente in un periodo di tempo per raccogliere i dati sulle prestazioni del canale:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

Informazioni su questa attività

Confermare che non vi sono messaggi non sottoposti a commit nella coda di trasmissione, come descritto in [“Verifica che i messaggi sulla coda siano disponibili”](#) a pagina 311, quindi controllare il campo XQTIME nell'output del comando di visualizzazione dello stato del canale. Quando i valori degli indicatori XQTIME sono costantemente elevati, o aumentano durante il periodo di misurazione, l'indicazione è che il canale non è al passo con le applicazioni di inserimento.

Eeguire i seguenti test:

Procedura

1. Verificare se le uscite sono in elaborazione.

Se le uscite vengono utilizzate sul canale che sta consegnando questi messaggi, potrebbero aggiungersi al tempo impiegato per l'elaborazione dei messaggi. Per identificare se questo è il caso, effettuare le seguenti verifiche:

- a) Nell'output del comando `DIS CHSTATUS(QM1.TO.QM2) ALL`, controllare il campo `EXITTIME`.
Se il tempo impiegato nelle uscite è superiore al previsto, esaminare l'elaborazione nelle uscite per eventuali loop non necessari o elaborazioni supplementari, in particolare nelle uscite di messaggi, di invio e di ricezione. Tale elaborazione influisce su tutti i messaggi spostati nel canale.
- b) Nell'output del comando `DIS CHSTATUS(QM1.TO.QM2) ALL`, controllare il campo `SUBSTATE`.
Se il canale ha una delle seguenti sottostazioni per un periodo di tempo significativo, esaminare l'elaborazione nelle uscite:
 - SCYEXIT
 - RCVEXIT
 - SENDEXIT
 - MSGEXIT
 - MREXIT

Per ulteriori informazioni sulle sottostazioni del canale, consultare la tabella [Tabella 32](#) a pagina 317.

2. Verificare se la rete è lenta.

Se i messaggi non si spostano abbastanza velocemente attraverso un canale, è possibile che la rete sia lenta. Per identificare se questo è il caso, effettuare le seguenti verifiche:

- a) Nell'output del comando `DIS CHSTATUS(QM1.TO.QM2) ALL`, controllare il campo `NETTIME`.
Questi indicatori vengono misurati quando il canale di invio richiede una risposta al partner. Ciò si verifica alla fine di ogni batch e quando un canale è inattivo durante l'heartbeat.
- b) Se questo indicatore mostra che i round trip impiegano più tempo del previsto, utilizzare altri strumenti di monitoraggio della rete per analizzare le prestazioni della rete.

3. Verificare se il canale sta utilizzando la compressione.

Se il canale sta utilizzando la compressione, ciò si aggiunge al tempo impiegato per l'elaborazione dei messaggi. Se il canale utilizza un solo algoritmo di compressione, effettuare le seguenti verifiche:

- a) Nell'output del comando `DIS CHSTATUS(QM1.TO.QM2) ALL`, controllare il campo `COMPTIME`.
Questi indicatori mostrano il tempo impiegato durante la compressione o la decompressione.
- b) Se la compressione scelta non riduce la quantità di dati da inviare della quantità prevista, modificare l'algoritmo di compressione.

4. Se il canale utilizza più algoritmi di compressione, effettuare le seguenti verifiche:
 - a) Nell'output del comando `DIS CHSTATUS(QM1.TO.QM2) ALL`, controllare i campi `COMPTIME`, `COMPHDR` e `COMPMSG`.
 - b) Modificare gli algoritmi di compressione specificati nella definizione del canale o considerare la scrittura di un'uscita del messaggio per sovrascrivere l'algoritmo di compressione del canale per determinati messaggi se la velocità di compressione o la scelta dell'algoritmo non fornisce la compressione o le prestazioni richieste.

Risoluzione dei problemi con i canali cluster

Se si dispone di un build di messaggi sul SISTEMA `SYSTEM.CLUSTER.TRANSMIT.QUEUE` queue, il primo passo nella diagnosi del problema consiste nel rilevare quale canale o canali stanno riscontrando un problema nella consegna dei messaggi.

Informazioni su questa attività

Per rilevare quale canale o canali, utilizzando `SYSTEM.CLUSTER.TRANSMIT.QUEUE` sta avendo un problema di consegna dei messaggi. Eseguire le seguenti verifiche:

Procedura

1. Emetti il seguente comando:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Nota: Se si dispone di un cluster occupato con molti messaggi in movimento, si consiglia di immettere questo comando con un numero maggiore per eliminare i canali che hanno solo pochi messaggi disponibili per la consegna.

2. Esaminare l'output per il canale o i canali che hanno valori elevati nel campo `XQMSGSA`. Determinare il motivo per cui il canale non sta spostando i messaggi o non li sta spostando abbastanza velocemente. Utilizzare le attività descritte in [“Monitoraggio dei canali”](#) a pagina 313 per diagnosticare i problemi con i canali che causano la creazione.

Monitoraggio dei cluster

All'interno di un cluster è possibile monitorare i messaggi dell'applicazione, i messaggi di controllo e i log. Esistono considerazioni di monitoraggio speciali quando il cluster esegue il bilanciamento del carico tra due o più istanze di una coda.

Monitoraggio dei messaggi dell'applicazione nel cluster

Generalmente, tutti i messaggi cluster che lasciano il gestore code passano attraverso `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, indipendentemente dal canale mittente del cluster utilizzato per trasmettere il messaggio. Ogni canale sta svuotando i messaggi destinati a tale canale in parallelo con tutti gli altri canali mittente del cluster. Un crescente accumulo di messaggi su questa coda può indicare un problema con uno o più canali e deve essere analizzato:

- La profondità della coda deve essere monitorata in modo appropriato per la progettazione del cluster.
- Il seguente comando restituisce tutti i canali che hanno più di un messaggio in attesa sulla coda di trasmissione:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Con tutti i messaggi cluster su una singola coda, non sempre è facile vedere quale canale ha problemi quando inizia a riempirsi. L'utilizzo di questo comando è un modo semplice per vedere quale canale è responsabile.

È possibile configurare un gestore code cluster in modo che abbia più code di trasmissione. Se si modifica l'attributo del gestore code `DEFCLXQ` in `CHANNEL`, ogni canale mittente del cluster viene associato a

una coda di trasmissione del cluster differente. In alternativa, è possibile configurare manualmente code di trasmissione separate. Per visualizzare tutte le code di trasmissione del cluster associate ai canali mittenti del cluster, eseguire il comando:

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

Definire le code di trasmissione cluster in modo che seguano il modello di avere la radice fissa del nome della coda sulla sinistra. È quindi possibile interrogare la profondità di tutte le code di trasmissione cluster restituite dal comando **DISPLAY CLUSMGR**, utilizzando un nome coda generico:

```
DISPLAY QUEUE (qname *) CURDEPTH
```

Monitoraggio dei messaggi di controllo nel cluster

La coda SYSTEM.CLUSTER.COMMAND.QUEUE viene utilizzata per elaborare tutti i messaggi di controllo cluster per un gestore code, generati dal gestore code locale o inviati a questo gestore code da altri gestori code nel cluster. Quando un gestore code mantiene correttamente il proprio stato cluster, questa coda tende a zero. Ci sono situazioni in cui la profondità dei messaggi su questa coda può temporaneamente aumentare:

- Avere molti messaggi nella coda indica il tasso di abbandono nello stato del cluster.
- Quando si apportano modifiche significative, consentire alla coda di stabilirsi tra tali modifiche. Ad esempio, quando si spostano i repository, consentire alla coda di raggiungere lo zero prima di spostare il secondo repository.

Mentre un backlog di messaggi è presente su questa coda, gli aggiornamenti allo stato del cluster o i comandi relativi al cluster non vengono elaborati. Se i messaggi non vengono rimossi da questa coda per un lungo periodo di tempo, è necessaria un'ulteriore analisi, inizialmente mediante l'ispezione dei log di errori del gestore code (o CHINIT log su z/OS) che potrebbe spiegare il processo che sta causando questa situazione.

Il SYSTEM.CLUSTER.REPOSITORY.QUEUE contiene le informazioni sulla cache del repository del cluster come un numero di messaggi. È consuetudine che i messaggi siano sempre presenti in questa coda e più per i cluster più grandi. Pertanto, la profondità dei messaggi su questa coda non è un problema.

Log di monitoraggio

I problemi che si verificano nel cluster potrebbero non mostrare sintomi esterni alle applicazioni per molti giorni (e anche mesi) dopo che il problema si verifica originariamente a causa della memorizzazione nella cache delle informazioni e della natura distribuita del clustering. Tuttavia, il problema originale viene spesso riportato nei log di errori IBM MQ (e nei log CHINIT su z/OS). Per questo motivo, è fondamentale monitorare attivamente questi log per tutti i messaggi scritti relativi al clustering. Questi messaggi devono essere letti e compresi, con qualsiasi azione intrapresa ove necessario.

Ad esempio: un'interruzione nelle comunicazioni con un gestore code in un cluster può determinare la conoscenza di alcune risorse cluster che vengono eliminate a causa del modo in cui i cluster riconvalidano regolarmente le risorse cluster ripubblicando le informazioni. Un'avvertenza di tale evento potenzialmente verificatosi viene riportata dal messaggio [AMQ9465](#) o [CSQX465I](#) sui sistemi z/OS. Questo messaggio indica che è necessario esaminare il problema.

Considerazioni speciali per il bilanciamento del carico

Quando il cluster esegue il bilanciamento del carico tra due o più istanze di una coda, le applicazioni che utilizzano devono elaborare i messaggi su ciascuna delle istanze. Se una o più di queste applicazioni terminano o arrestano l'elaborazione dei messaggi, è possibile che il clustering continui a inviare messaggi a tali istanze della coda. In questa situazione, i messaggi non vengono elaborati fino a quando le applicazioni non funzionano nuovamente correttamente. Per questo motivo, il controllo delle applicazioni è una parte importante della soluzione e occorre intraprendere azioni per reinstradare i messaggi in tale

situazione. Un esempio di meccanismo per automatizzare tale monitoraggio è il seguente: [Programma di esempio Monitoraggio coda cluster \(AMQSCLM\)](#).

Concetti correlati

“Ottimizzazione delle reti di pubblicazione / sottoscrizione distribuite” a pagina 383

Utilizzare i suggerimenti di ottimizzazione in questa sezione per migliorare le prestazioni dei cluster e delle gerarchie di pubblicazione / sottoscrizione distribuiti IBM MQ .

“Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione” a pagina 389

Un concetto importante nelle prestazioni di messaggistica asincrona è *balance*. A meno che i destinatari dei messaggi non siano bilanciati con i produttori di messaggi, esiste il rischio che un backlog di messaggi non utilizzati possa accumularsi e compromettere seriamente le prestazioni di più applicazioni.

Monitoraggio della commutazione della coda di trasmissione

È importante monitorare il processo delle code di trasmissione di commutazione dei canali mittente del cluster in modo da ridurre al minimo l'impatto sull'azienda. Ad esempio, non tentare questo processo quando il carico di lavoro è elevato o commutando molti canali contemporaneamente.

Il processo di commutazione dei canali

Il processo utilizzato per cambiare canale è:

1. Il canale apre la nuova coda di trasmissione per l'input e inizia a ricevere i messaggi da esso (utilizzando l'ID di correlazione get by)
2. Un processo in background viene avviato dal gestore code per spostare i messaggi accodati per il canale dalla vecchia coda di trasmissione alla nuova coda di trasmissione. Mentre i messaggi vengono spostati, tutti i nuovi messaggi per il canale vengono accodati alla vecchia coda di trasmissione per preservare la sequenza. Il completamento di questo processo potrebbe richiedere del tempo se è presente un numero elevato di messaggi per il canale sulla vecchia coda di trasmissione o se i nuovi messaggi stanno rapidamente arrivando.
3. Quando nessun messaggio di cui è stato eseguito il commit o di cui non è stato eseguito il commit rimane in coda per il canale nella sua vecchia coda di trasmissione, la commutazione viene completata. I nuovi messaggi vengono ora inseriti nella nuova coda di trasmissione.

Per evitare l'eventualità di una commutazione simultanea di numerosi canali, IBM MQ fornisce la possibilità di commutare la coda di trasmissione di uno o più canali che non sono in esecuzione. Su:

- IBM MQ for Multiplatforms il comando viene denominato **runswch1**
- IBM MQ for z/OS il programma di utilità CSQUTIL può essere utilizzato per elaborare un comando SWITCH CHANNEL

Monitoraggio dello stato delle operazioni di switch

Per comprendere lo stato delle operazioni di switch, gli amministratori possono eseguire le azioni riportate di seguito:

- Monitorare il log degli errori del gestore code (AMQERR01.LOG) in cui i messaggi vengono emessi per indicare le seguenti fasi durante l'operazione:
 - L'operazione di commutazione è stata avviata
 - Lo spostamento dei messaggi è stato avviato
 - Aggiornamenti periodici sul numero di messaggi da spostare (se l'operazione di commutazione non viene completata rapidamente)
 - Lo spostamento dei messaggi è stato completato
 - L'operazione di commutazione è stata completata

Su z/OS, questi messaggi vengono emessi nel log del lavoro del gestore code, non nel log del lavoro dell'iniziatore del canale, anche se un singolo messaggio viene emesso da un canale nel log del lavoro dell'iniziatore del canale se viene avviato un commutatore all'avvio.

- Utilizzare il comando `DISPLAY CLUSQMGR` per interrogare la coda di trasmissione attualmente utilizzata da ciascun canale mittente del cluster.
- Eseguire il comando `runswch1` (o `CSQUTIL` su z/OS) in modalità query per verificare lo stato di commutazione di uno o più canali. L'output di questo comando identifica quanto segue per ciascun canale:
 - Se il canale ha un'operazione di commutazione in sospeso
 - Quale coda di trasmissione il canale sta passando da e a
 - Il numero di messaggi rimasti nella vecchia coda di trasmissione

Ogni comando è davvero utile, perché in una chiamata è possibile determinare lo stato di ogni canale, l'impatto che una modifica della configurazione ha avuto e se tutte le operazioni di switch sono state completate.

Potenziali problemi che potrebbero verificarsi

Consultare [Problemi potenziali durante la commutazione delle code di trasmissione](#) per un elenco di alcuni problemi che potrebbero verificarsi durante la commutazione della coda di trasmissione, le relative cause e le soluzioni più probabili.

Concetti correlati

[“Ottimizzazione delle reti di pubblicazione / sottoscrizione distribuite”](#) a pagina 383

Utilizzare i suggerimenti di ottimizzazione in questa sezione per migliorare le prestazioni dei cluster e delle gerarchie di pubblicazione / sottoscrizione distribuiti IBM MQ .

[“Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione”](#) a pagina 389

Un concetto importante nelle prestazioni di messaggistica asincrona è *balance*. A meno che i destinatari dei messaggi non siano bilanciati con i produttori di messaggi, esiste il rischio che un backlog di messaggi non utilizzati possa accumularsi e compromettere seriamente le prestazioni di più applicazioni.

Multi

Monitoraggio del bilanciamento dell'applicazione

È possibile utilizzare il comando `DISPLAY APSTATUS` per monitorare lo stato del bilanciamento dell'applicazione in un cluster uniforme e per determinare il motivo per cui l'applicazione non è bilanciata se ciò non è previsto.

Monitoraggio dello stato corrente delle applicazioni tra i gestori code in un cluster

Da qualsiasi gestore code in un cluster uniforme, è possibile ottenere una panoramica dello stato corrente delle applicazioni in tutti i gestori code di un cluster eseguendo il comando `DIS APSTATUS` .

Da IBM MQ 9.2.0 il campo **TYPE** viene visualizzato anche nell'output.

Ad esempio, dopo che un gestore code è stato appena avviato, potrebbe essere visualizzato un output simile al seguente:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)
```

Ciò dimostra che esiste un'applicazione denominata MYAPP nel cluster uniforme, e attualmente ci sono otto istanze, tutte considerate mobili intorno al cluster uniforme. Il valore bilanciato di UNKNOWN è un valore temporaneo, che indica che il gestore code non ha ancora tentato di ribilanciare l'applicazione, se necessario.

Dopo un breve periodo di tempo, è più probabile che si visualizzi il seguente output:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                       MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

Questo output mostra che l'applicazione ha otto istanze, ma che non sono bilanciate nel cluster uniforme. A questo punto, vale la pena esaminare la distribuzione delle applicazioni nel cluster.

A tale scopo, eseguire nuovamente il comando **DIS APSTATUS** . Notare che è possibile eseguire questo comando su qualsiasi gestore code nel cluster 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)
```

Da questo, è possibile vedere a questo punto che il gestore code UNID001 ha sei istanze, ma i gestori code UNID0002 e UNID0003 ne hanno solo una. L'output BALSTATE fornisce un'indicazione dello stato di bilanciamento l'ultima volta che il gestore code lo ha segnalato. Tuttavia, tenere presente che il conteggio delle istanze potrebbe essere più recente del campo BALSTATE.

Questo output fornisce inoltre una buona indicazione che il cluster uniforme sta distribuendo informazioni su questa applicazione. L'applicazione è nota a tutte le istanze del gestore code nel cluster uniforme e la data e ora dell'ultimo messaggio è molto recente.

Inoltre, il campo ACTIVE indica che tutti i gestori code nel cluster sono considerati in comunicazione tra loro; se ACTIVE è impostato su NO su qualsiasi gestore code, indica che la comunicazione regolare con esso è stata interrotta.

Se lasciato a riequilibrare se stesso, l'output finirebbe per essere qualcosa come:

```
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)                       MOVCCOUNT(2)
BALSTATE(OK)                    LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)              QMNAME(UNID003)
```

```
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
```

BALSTATE dei gestori code è ora OK per indicare che è stato raggiunto uno stato stazionario.

Monitoraggio delle singole istanze dell'applicazione

È possibile esaminare le singole istanze dell'applicazione, ma ciò viene fatto per ciascun gestore code. Guardando l'output su 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 presenza di MOVABLE (YES) indica che questa istanza può essere spostata, se necessario, in un altro gestore code nel cluster. Nel seguente esempio, l'istanza di applicazione non è spostabile perché non si connette come client:

```
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)
```

È possibile utilizzare il campo **CONNTAG** per visualizzare le singole connessioni del gestore code da tale istanza, il che può essere utile se l'istanza dell'applicazione è IMMDIMINN (NOTRECONN). Nel seguente esempio, l'applicazione client non è spostabile e l'analisi delle opzioni di connessione mostra che non dispone di MQCNO_RECONNECT nel 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)                               TID(20)
  APPLDESC(IBM MQ Channel)                APPLTAG(ClientApp)
  APPLTYPE(USER)                          ASTATE(NONE)
  CHANNEL(SYSTEM.DEF.SVRCONN)             CLIENTID( )
  CONNAME(127.0.0.1)
```

```
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserId) UOWLOG( )
UOWSTDA( ) UOWSTTI( )
UOWLOGDA( ) UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0) UOWSTATE(NONE)
CONN TAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
TYPE (CONN)
```

Concetti correlati

[Bilanciamento automatico dell'applicazione](#)

z/OS

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](#)

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 331](#) 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 328](#)
- z/OS service aids:
 - [“Using System Management Facility” on page 329](#)
- 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 332](#).

Information about interpreting the data gathered by the accounting trace is given in [“Interpreting IBM MQ for z/OS accounting data” on page 369](#).

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 329.

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 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 328 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 “Monitoraggio della tua rete 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 “Tipi di evento” 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 331.

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 369](#).

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 337](#)).

Utilizzare questo argomento come riferimento per i comandi necessari per raccogliere record SMF di tipo 115 e di tipo 116.

Utilizzo del comando START TRACE

Utilizzare la versione dinamica del comando START TRACE con le seguenti opzioni per raccogliere i record:

- START TRACE(STAT) DEST SMF CLASS(*) e START TRACE(ACCTG) DEST SMF CLASS(*) avviano le tracce per le classi da 1 a 3
- START TRACE(STAT) DEST SMF CLASS(4) e START TRACE(ACCTG) DEST SMF CLASS(4) avviano rispettivamente le statistiche dell'iniziatore di canali e i dati di account di canale.

Consultare Pianificazione dei dati SMF dell'iniziatore di canali per ulteriori dettagli sulle informazioni dell'iniziatore di canali di classe 4.

- START TRACE(STAT) DEST SMF CLASS(5) avvia le statistiche della coda.



Attenzione: Se si sta utilizzando la versione console del comando, aggiungere cpf all'inizio del comando immesso.

Per le statistiche del canale CLASS (4), è necessario impostare l'attributo STATCHL sulla definizione CHANNEL. Per ulteriori informazioni, consultare “Controllo della raccolta delle statistiche del canale” a pagina 147.

Per le statistiche della coda CLASS (5), è necessario impostare l'attributo STATQ sulla definizione QUEUE e / o QMGR. Per ulteriori informazioni, consultare “Controllo della raccolta delle statistiche della coda” a pagina 146.

Record di tipo 115 SMF - Statistiche IBM MQ

Per verificare la raccolta delle statistiche correnti, immettere il comando DISPLAY TRACE DISPLAY TRACE (STAT). Dovresti vedere quanto segue:

```
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: L'impostazione dell'attributo ZPARM SMFSTAT o ZPARM SMFACCT su un asterisco ora controlla solo le classi 1, 2 e 3. È necessario attivare le classi 4 e 5 utilizzando il comando START TRACE.

Dati SMF di tipo 116 Classe 3 & 4 - IBM MQ record di account di canale e attività

Per verificare la raccolta delle statistiche correnti, immettere il comando DISPLAY TRACE DISPLAY TRACE (ACCT). Dovresti vedere quanto segue:

```
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
```

È possibile attivare e disattivare dinamicamente la raccolta dati SMF116 poiché non è necessario alcun ciclo del gestore code, utilizzando i seguenti comandi:

```
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)
```

I dati di contabilità del canale richiedono un'ulteriore configurazione per produrre lo SMF. I passi sono:

- Immettere il comando per attivare la raccolta delle informazioni per tutte le connessioni client e destinatario del mittente:

```
ALTER QMGR STATCHL(HIGH)
```

- Immettere il comando per attivare la raccolta delle informazioni per i canali cluster definiti automaticamente se il cluster è in uso:

```
ALTER QMGR STATCLS(HIGH)
```

Nota: Nelle istruzioni precedenti, il valore di HIGH, MED o LOW ha lo stesso effetto.

- Verificare che tutti i canali abbiano STATCHL impostato su QMGR:

```
DISPLAY CHANNEL(*) STATCHL
```

Verifica di STATime



Attenzione: È necessario impostare l'intervallo in modo che non sia superiore a cinque minuti per essere conformi agli strumenti più recenti, utilizzando il comando [SET SYSTEM](#), anche se il valore predefinito è 30 minuti.

Se si sta valutando un gruppo di condivisione code (QSG), assicurarsi che STATIME sia congruente in tutto il QSG.

- Visualizzare le impostazioni di sistema utilizzando il seguente comando [DISPLAY SYSTEM](#):

```
DISPLAY SYSTEM
```

- Se il valore STATIME è zero, che indica che l'intervallo delle statistiche è impostato sul valore LPAR predefinito, di solito è OK.

Ci sono stati casi di LPAR di volumi molto elevati che hanno creato durate variabili della produzione di dati SMF. Se nella valutazione dei dati sono state riscontrate durate molto diverse, impostare STATIME su un valore diverso da zero per tutti i gestori code.

- I seguenti esempi impostano l'intervallo STATIME su cinque minuti:

```
SET SYSTEM STATIME(05)
```

o se in IBM MQ 9.2.4 o superiore:

```
SET SYSTEM STATIME(05.00)
```

Nota: Il nuovo intervallo non diventa effettivo fino a quando non è trascorso l'intervallo corrente, quindi è necessario apportare la modifica prima dell'inizio della raccolta dati.

Verifica del valore ACCTIME per i gestori code al livello di rilascio IBM MQ 9.2.4 o superiore:

- Se ACCTIME è impostato su -1, è uguale all'attributo STATIME.
- Se questa valutazione è per un QSG, assicurarsi che ACCTIME sia congruente in tutto il QSG. In caso contrario, impostare i valori sullo stesso intervallo prima di iniziare a raccogliere i dati.
- Se ACCTIME è impostato su 30 o superiore, modificarlo in modo che non sia maggiore di 15 utilizzando il comando [SET SYSTEM](#). Gli esempi seguenti impostano l'intervallo ACCTIME su cinque minuti:

```
SET SYSTEM ACCTIME(05.00)
```

Nota: Il nuovo intervallo non diventa effettivo fino a quando non è trascorso l'intervallo corrente, quindi è necessario apportare la modifica prima dell'inizio della raccolta dati.

Verifica dell'impostazione ACCTQ sui gestori code

- Visualizzare l'impostazione ACCTQ per il gestore code che si sta analizzando utilizzando il seguente comando:

```
DISPLAY QMGR ACCTQ
```

- Se il valore è ACCTQ (ON), non è necessaria alcuna ulteriore azione. Altrimenti, immettere il seguente comando:

```
ALTER QMGR ACCTQ(ON)
```

- Se il clustering è in uso, assicurarsi che l'account sia attivato per il SISTEMA SYSTEM.CLUSTER.TRANSMIT.QUEUE e qualsiasi altra coda di trasmissione cluster denominata ospitata dal gestore code. Per determinare se l'account è abilitato per le code di trasmissione del cluster:
 - Visualizzare l'impostazione ACCTQ per la coda o le code, utilizzando il seguente comando:

```
DISPLAY QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ
```

- Se il valore è ACCTQ (ON), non è necessaria alcuna ulteriore azione. Altrimenti, immettere il seguente comando:

```
ALTER QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ(ON)
```

- Per tutte le altre code di volumi elevati, verificare che il valore di ACCTQ sia ON o impostare ACCTG su 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 337](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 338](#)

Use this topic to understand some example SMF records.

The SMF header

Use this topic as a reference for the format of the SMF header.

[Table 34 on page 337](#) 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.	

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 337](#) 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
Log manager	1	116	X'74'	QJST
Message manager	2	36	X'24'	QMST
Data manager	2	44	X'2C'	QIST

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	
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 338.

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 339 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.

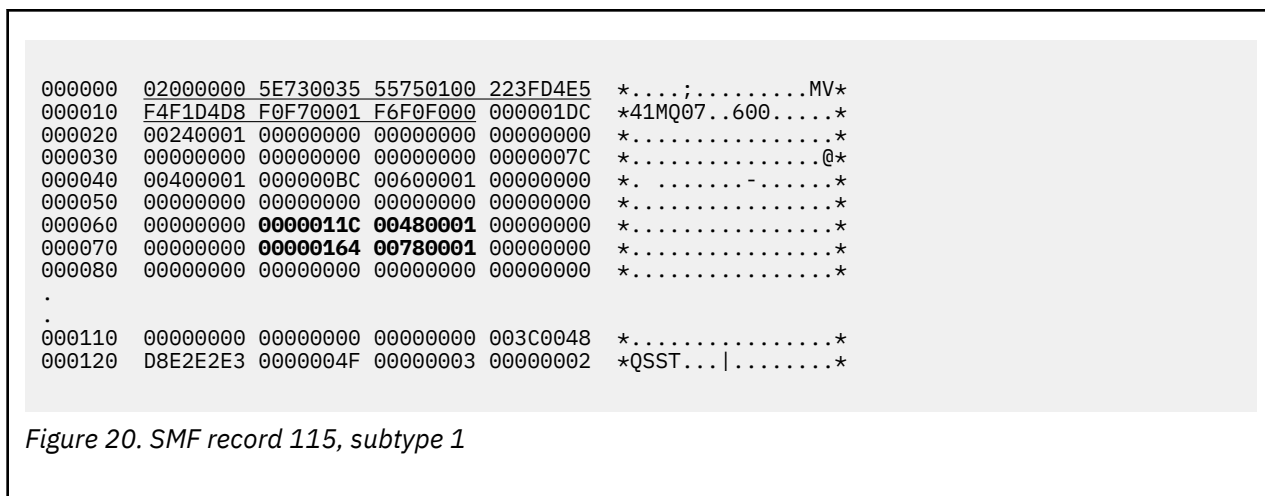


Figure 21 on page 340 shows an example of part of the SMF record for subtype 2. Subtype 2 includes the statistics records for the message, data, lock, coupling facility, topic, and Db2 managers. The SMF record header is shown underlined; the self-defining sections are shown alternately **bold** and *italic*.

- 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 IFASMFDL 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 329.

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(CSQDQSRs).

Record di dati del gestore log

Utilizzare questo argomento come riferimento per il formato dei dati del gestore log.

Il formato del record delle statistiche del gestore log è descritto nella macro assembler thlqual.SCSQMACS(CSQDQJST).

Nelle statistiche, questi conteggi sono importanti:

1. Il numero totale di richieste di scrittura log:

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. Il numero totale di richieste di letture log:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

I problemi che possono essere esaminati utilizzando le statistiche del gestore log sono descritti nella seguente tabella.

<p>Sintomo 1 QJSTWTB è diverso da zero.</p> <p>Motivo Le attività vengono sospese mentre il buffer in memoria viene scritto nel log attivo. Potrebbero essersi verificati dei problemi durante la scrittura nel log attivo. Il parametro OUTBUFF in CSQ6LOGP è troppo piccolo.</p> <p>Azione Esaminare i problemi di scrittura nel log attivo. Aumentare il valore del parametro OUTBUFF in CSQ6LOGP.</p>
<p>Sintomo 2 Il rapporto: $\text{QJSTWTL} / N_{\text{logread}}$ è maggiore dell' 1%.</p> <p>Motivo Sono state avviate letture di log che dovevano essere lette da un log di archivio, ma IBM MQ non è stato in grado di assegnare un dataset perché i dataset MAXRTU erano già assegnati.</p> <p>Azione Aumentare MAXRTU.</p>

Sintomo 3

Il rapporto: $QJSTRARH/N_{logread}$ è maggiore del normale.

Motivo

La maggior parte delle richieste di lettura del log deve provenire dal buffer di output o dal log attivo. Per soddisfare le richieste di backout, i record dell'unità di recupero vengono letti dal buffer in memoria, dal log attivo e dai log archiviati.

Un'unità di ripristino di lunga durata, che si estende su un periodo di molti minuti, potrebbe avere record di log distribuiti su molti log differenti. Ciò riduce le prestazioni perché è necessario eseguire un lavoro supplementare per ripristinare i record di log.

Azione

Modificare l'applicazione per ridurre la lunghezza di un'unità di ripristino. Inoltre, si consiglia di aumentare la dimensione del log attivo per ridurre la possibilità che una singola unità di ripristino venga distribuita su più di un log.

Altri puntatori

Il rapporto $N_{logread} / N_{logwrite}$ fornisce un'indicazione di quanto lavoro deve essere ripristinato.

Sintomo 4

$QJSTLLCP$ è più di 10 all'ora.

Motivo

Su un sistema occupato, ci si aspetterebbe di vedere in genere 10 checkpoint all'ora. Se il valore di $QJSTLLCP$ è maggiore di questo valore, indica un problema nella configurazione del gestore code.

Il motivo più probabile è che il parametro $LOGLOAD$ in $CSQ6SYSP$ è troppo piccolo. L'altro evento che causa un punto di controllo è quando un log attivo si riempie e passa al successivo dataset di log attivo. Se i tuoi log sono troppo piccoli, questo può causare checkpoint frequenti.

$QJSTLLCP$ è un conteggio del numero totale di checkpoint.

Azione

Aumentare il parametro $LOGLOAD$ o aumentare la dimensione dei dataset di log come richiesto.

Sintomo 5

$QJSTCmpFail > 0$ o $QJSTCmpComp$ non meno di $QJSTCmpUncmp$

Motivo

Il gestore code non è in grado di comprimere in modo significativo i record di log.

$QJSTCmpFail$ è il numero di volte in cui il gestore code non è stato in grado di ottenere alcuna riduzione della lunghezza del record. È necessario confrontare il numero con $QJSTCmpReq$ (numero di richieste di compressione) per vedere se il numero di errori è significativo.

$QJSTCmpComp$ è il totale dei byte compressi scritti nel log e $QJSTCmpUncmp$ è il numero totale di byte prima della compressione. Nessuno dei due contiene byte scritti per i record di log che non erano idonei per la compressione. Se i numeri sono simili, la compressione ha ottenuto pochi benefici.

Azione

Disattivare la compressione log. Immettere il comando `SET LOG COMPLOG (NONE)`. Per i dettagli, consultare il comando `SET LOG`.

Nota: Nella prima serie di statistiche prodotte dopo l'avvio del sistema, potrebbe essere presente un'attività di log significativa a causa della risoluzione delle unità di ripristino inutilizzate.

Modifiche alla scrittura zHyper

V 9.4.0

Da IBM MQ 9.3.5 in poi, l'elaborazione zHyperWrite modifica il comportamento delle statistiche $QJSTHWC$ e $QJSTHWE$.

Prima di IBM MQ 9.3.5:

- QJSTHWC è il numero di dataset di log utilizzati nell'intervallo SMF che si trovano su volumi con capacità di scrittura zHyper. La capacità di scrittura zHyperdi un volume si ottiene all'avvio del gestore code e può cambiare nel tempo, quindi queste informazioni potrebbero essere obsolete.
- QJSTHWE è il numero di dataset di log utilizzati nell'intervallo SMF scritti con la scrittura di zHyperabilitata. Ciò si verifica se il data set del log si trova su un volume con capacità di scrittura zHyper se il gestore code è stato abilitato per la scrittura zHyperimpostando ZHYWRITE (YES).

Da IBM MQ 9.3.5:

- QJSTHWC è il numero di dataset di log utilizzati nell'intervallo SMF che si trovano su volumi con capacità di scrittura zHyper. La capacità di scrittura zHyperdi un volume si ottiene all'avvio del gestore code e può cambiare nel tempo, quindi queste informazioni potrebbero essere obsolete.
- QJSTHWE è il numero di dataset di log utilizzati nell'intervallo SMF scritti con la scrittura di zHyperabilitata. Ciò si verifica se il gestore code è stato abilitato per la scrittura zHyperimpostando ZHYWRITE (YES).



Attenzione: La capacità di scrittura zHyperdi un volume può cambiare nel tempo. Da IBM MQ 9.3.5 ciò può risultare in scenari in cui QJSTHWE è maggiore di QJSTHWC.

Statistiche zHyperLink

V 9.4.0

Da IBM MQ 9.4.0 in poi, le statistiche del gestore log sono state aggiunte per tenere traccia delle prestazioni di zHyperLink.

Le seguenti statistiche sono state aggiunte al QJST:

Nome	Tipo	Byte di lunghezza	Descrizione
DJSTHLSCIW	Numero intero senza segno	4	Numero di scritture a pagina singola che hanno richiesto zHyperLink.
QJSTHLMCIW	Numero intero senza segno	4	Numero di scritture di più pagine che hanno richiesto il link zHyper.
QJSTHLCICNTW	Numero intero senza segno	4	Numero di pagine scritte dove la scrittura ha richiesto il link zHyper.
QJSTHLSCIS	Numero intero senza segno	4	Numero di scritture pagina che hanno utilizzato correttamente il link zHyper. Cioè, la scrittura è stata effettuata in modo sincrono.
QJSTHLMCIS	Numero intero senza segno	4	Numero di scritture di più pagine che hanno utilizzato correttamente zHyperLink. Cioè, la scrittura è stata effettuata in modo sincrono.
QJSTHLCICNTS	Numero intero senza segno	4	Numero di pagine scritte correttamente utilizzando il collegamento zHyper. Cioè, la scrittura è stata effettuata in modo sincrono.
QJSTHLSCIF	Numero intero senza segno	4	Il numero di scritture di pagine singole che hanno tentato di utilizzare zHyperLink, ma non è stato possibile eseguire la scrittura in modo sincrono. Invece, la scrittura è stata eseguita correttamente in modo asincrono.

Nome	Tipo	Byte di lunghezza	Descrizione
QJSTHLMCIF	Numero intero senza segno	4	Numero di scritture di più pagine che hanno tentato di utilizzare zHyperLink, ma non è stato possibile eseguire la scrittura in modo sincrono. Invece, la scrittura è stata eseguita correttamente in modo asincrono.
CICNTF QJSTHLNTF	Numero intero senza segno	4	Numero di pagine scritte in cui la scrittura ha tentato di utilizzare zHyperLink, ma non è stato possibile eseguire la scrittura in modo sincrono. Invece, la scrittura è stata eseguita correttamente in modo asincrono.
QJSTHLSCICON CNT	Numero intero senza segno	4	Il numero di scritture pagina singola prima che fosse stabilita una sessione di scrittura link zHyper. Consultare la nota "1" a pagina 344.
CCNT QJSTHLMCICON T	Numero intero senza segno	4	Il numero di scritture di più pagine prima che fosse stabilita una sessione di scrittura link zHyper. Consultare la nota "1" a pagina 344.
QJSTHLCONCIC NT	Numero intero senza segno	4	Numero di pagine scritte prima che fosse stabilita una connessione zHyperLink. Consultare la nota "1" a pagina 344.
QJSTHLSCIWTM AX	Numero intero senza segno	8	Tempo di scrittura link zHyperpiù lungo nell'intervallo, pagina singola.
QJSTHLSCIWTM IN	Numero intero senza segno	8	Tempo di scrittura link zHyperpiù breve nell'intervallo, pagina singola.
QJSTHLSCIWTT OT	Numero intero senza segno	8	Tempo totale di scrittura zHyperLink nell'intervallo, pagina singola.
QJSTHLMCIWT MAX	Numero intero senza segno	8	Tempo di scrittura link zHyperpiù lungo nell'intervallo, più pagine.
QJSTHLMCIWT MIN	Numero intero senza segno	8	Tempo di scrittura link zHyperpiù breve nell'intervallo, più pagine.
QJSTHLMCIWTT OT	Numero intero senza segno	8	Tempo totale di scrittura zHyperLink nell'intervallo, più pagine.
QJSTHLIOSQU	Numero intero senza segno	16	Somma dei quadrati dei tempi di scrittura dei link zHyper, per le scritture di pagine singole.
QJSTHLC	Numero intero senza segno	4	Numero di nuovi log, utilizzati in questo intervallo SMF, che supportano zHyperLink.
QJSTHLE	Numero intero senza segno	4	Numero di nuovi log, utilizzati in questo intervallo SMF, che sono zHyperLink abilitati.

Nota:

1. Quando si passa a una nuova copia di log attiva, potrebbe essere necessario un periodo di tempo prima che venga stabilita una sessione di scrittura zHyperLink con DASD.

 **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.

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.

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 337](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 338](#)

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 347](#).

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
44	2C	Integer	4	LISTCNT	Number of list requests
48	30	Integer	4	READCNT	Number of read requests
52	34	Integer	4	UPDTCNT	Number of update requests
56	38	Integer	4	WRITCNT	Number of write requests
60	3C	Integer	4	SCSSEL	SCST (shared-channel-status) selects

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
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
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

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
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	SCSUTCW	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	SCSDTCW	Cumulative STCK difference - Thread delete
368	170	Character	8	SCSDTMXW	Maximum STCK difference - Thread delete
376	178	Character	8	SCSDSCW	Cumulative STCK difference - SQL delete
384	180	Character	8	SCSDSMXW	Maximum STCK difference - SQL delete
392	188	Character	8	SSKSTCW	Cumulative STCK difference - Thread select
400	190	Character	8	SSKSTMXW	Maximum STCK difference - Thread select
408	198	Character	8	SSKSSCW	Cumulative STCK difference - SQL select
416	1A0	Character	8	SSKSSMXW	Maximum STCK difference - SQL select
424	1A8	Character	8	SSKITCW	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	SSKDTCW	Cumulative STCK difference - Thread delete
464	1D0	Character	8	SSKDTMXW	Maximum STCK difference - Thread delete
472	1D8	Character	8	SSKDSCW	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
508	IFC	64 bit integer	8	LMSSTCW	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	LMSSSCW	Total elapsed time for all SQL read BLOB requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
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 RRSF 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
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

Table 37. Coupling facility statistics record (QEST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
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.

Table 38. Topic manager statistics record (QTST)

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	QTSTMSG	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)
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

Offset: Dec	Offset: Hex	Type	Len	Name	Description
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 `th1qual.SCSQMACS(CSQDQESD)` and in C header file `th1qual.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 `th1qual.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qual.SCSQMACS(CSQDQWSX)`.

Table 39 on page 354 shows the format of the 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
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)

Table 39. Structure of the channel initiator statistics self-defining section (continued)

Offset: Dec	Offset : Hex	Type	Length	Name	Description
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

Table 41. Key fields in the QWHS (continued)

Name	Length	Description
QWHSSMFC	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
QWHSDURN	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 357.
- 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 358
- [“Adapter tasks”](#) on page 359
- [“Domain Name Server \(DNS\) task”](#) on page 360
- [“SSL tasks”](#) on page 361

Each task includes:

- The elapsed time that the task spent processing requests in the interval (*qcteltm*)
- 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 (*qctcptm*)
- The total wait time of this task in the interval (*qctwttm*)
- The number of requests in the interval (*qctreqn*)

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 (*qctlgdu*, *qctlsdu*) and the time of day when this occurred (*qctlgdm*, *qctlsdm*) 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 (*qcteltm*) 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`.

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:

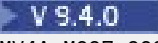
- CHINIT job name (`qcctjobn`)
- QSG name if it is in a queue sharing group (`qcctqsgn`)
- Peak number used of current channels (`qcctnocc`)
- Peak number used of active channels (`qcctnoac`)
- MAXCHL - maximum permitted current channels (`qcctmxcc`)
- ACTCHL - maximum permitted active channels (`qcctmxac`)
- TCPCHL - maximum permitted TCP/IP channels (`qcctmxtp`)
- LU62CHL - maximum permitted LU62 channels (`qcctmxlu`)
-  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.
-  64-bit storage limit available to the CHINIT (`qcctslim`)
-  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:

```
 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.

z/OS 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 `thlqual.SCSQC370 (CSQDSMFC)`, and in assembler macro `thlqual.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 `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.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, Seconds,	CPU %,	"avg CPU", uSeconds,	"avg ET", uSeconds,	longest, uSeconds,
0,	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,
12:46:40.237697								2014/03/18,
1,	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,
12:46:18.938022								2014/03/18,
2,	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,
12:46:35.358145								2014/03/18,
3,	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,
12:46:44.514045								2014/03/18,
4,	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,
12:46:22.134123								2014/03/18,
Summ,	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,
12:46:22.134123								2014/03/18,

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 %: *qcteltn* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltn* 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 (*qcteltn*) 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 370 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 372](#).

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 371](#) 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 373
0	Message manager	44	X'2C'	“Message manager data records” on page 375
1	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 376
1	Thread-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 376
1	Queue-level accounting	52	X'34'	“Thread-level and queue-level data records” on page 376 . 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 376
2	Queue-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 376
10	Channel accounting			“Channel accounting data records” on page 379

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

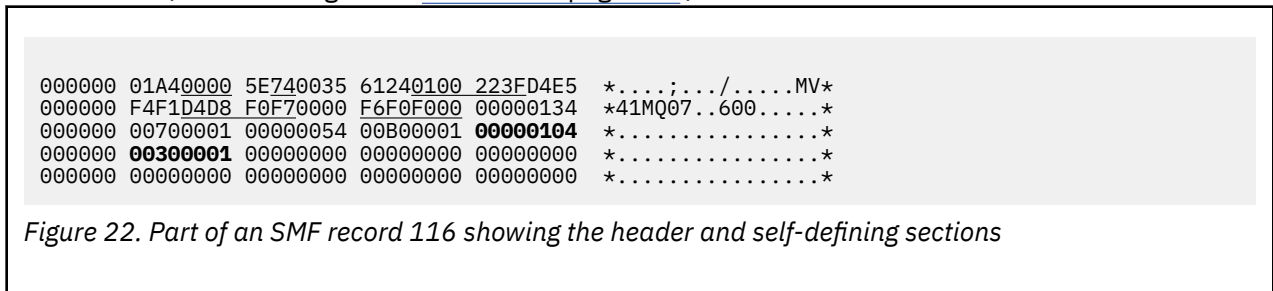
Halfword giving the length of this data record.

NNNN

Halfword giving the number of data records in this SMF record.

Figure 22 on page 372 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 370 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 371) 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.

z/OS 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 329](#).

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 373 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.

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	QWHSSID	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 374](#)
- IMS connections (QWHCATYP=3 or 4) - see [Table 46 on page 374](#)
- 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 375 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 377 and Figure 25 on page 377 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 377 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....*

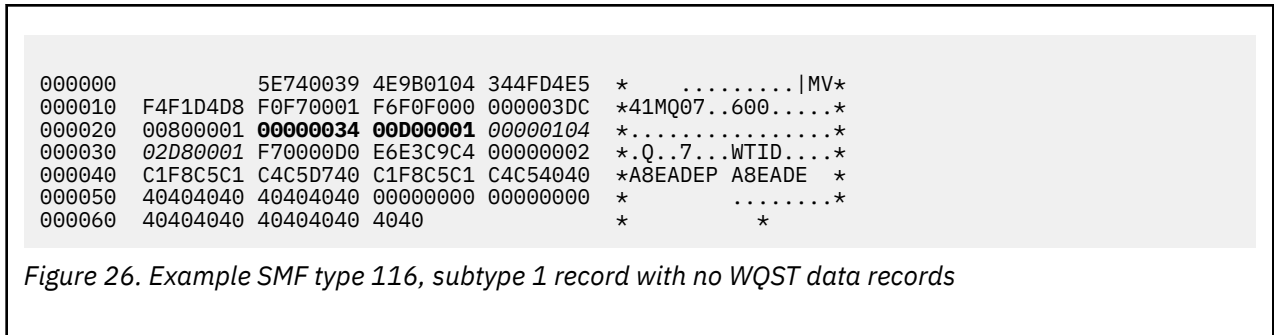
```

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 378 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 377.

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 378 shows the format of the self-defining section.

Table 48. Structure of the channel accounting 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 luddt`)
- Channel type (`qcstc hty`)
- Common name (CN) from SSLCERTI (`qcstc slcn`)
- 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

```

Ottimizzazione della rete IBM MQ

Utilizzare i suggerimenti per l'ottimizzazione contenuti in questa sezione per migliorare le prestazioni della rete del gestore code.

Ottimizzazione dei canali di connessione client e server

L'impostazione predefinita per **SHARECNV** è 10, che consente fino a 10 conversazioni client per ciascuna istanza del canale. Tuttavia, l'utilizzo di un diverso numero di conversazioni condivise può essere migliore per le prestazioni. Se non sono necessarie conversazioni condivise o se si utilizza un server distribuito, impostare **SHARECNV** su 1. Se si dispone di applicazioni client esistenti che non vengono eseguite correttamente quando si imposta **SHARECNV** su 1 o su un valore superiore, impostare **SHARECNV** su 0.

Informazioni su questa attività

Per alcune configurazioni, l'utilizzo di conversazioni condivise porta vantaggi significativi. Tuttavia, per i server distribuiti, l'elaborazione dei messaggi sui canali che utilizzano la configurazione predefinita di 10 conversazioni condivise è in media del 15% più lenta rispetto ai canali che non utilizzano le conversazioni condivise. Su un'istanza del canale MQI che condivide conversazioni, tutte le conversazioni su un socket vengono ricevute dallo stesso thread. Se le conversazioni che condividono un socket sono tutte occupate, i thread di conversazione si contendono l'un l'altro per utilizzare il thread di ricezione. La contesa causa ritardi e in questa situazione è meglio utilizzare un minor numero di conversazioni condivise.

Utilizzare il parametro **SHARECNV** per specificare il numero massimo di conversazioni da condividere su una particolare istanza del canale client TCP/IP. Per i dettagli di tutti i possibili valori, consultare [Supported IBM MQ client: Default behavior of client - connection and server - connection channels](#).

Se si imposta **SHARECNV** su 1 o superiore, si abilitano i seguenti miglioramenti delle prestazioni:

- Heartbeat bidirezionali
- Arresto o inattività amministratore
- Lettura anticipata
- Asincrono - utilizza dalle applicazioni client

Se non hai bisogno di conversazioni condivise, queste due impostazioni offrono le migliori prestazioni:

- **SHARECNV**(1).
- **SHARECNV**(0).

Note:

- Se il valore della connessione client **SHARECNV** non corrisponde a quello della connessione server **SHARECNV**, viene utilizzato il valore più basso.
- Quando le applicazioni vengono collegate o compilate rispetto a una libreria non rientrante, il valore CURSHCNV (0) viene negoziato anche se viene impostato un valore superiore in CLNTCONN e SVRCONN.

Per ottimizzare le prestazioni per una determinata istanza del canale, completare una delle seguenti operazioni.

Procedura

- Monitorare i canali che utilizzano il valore predefinito SHARECNV di 10.

L'impostazione predefinita di SHARECNV (10) funziona correttamente in molti scenari, ma potrebbe non essere l'impostazione ottimale per una determinata istanza del canale. Ad esempio, per i server distribuiti, l'elaborazione dei messaggi sui canali che utilizzano questa opzione è in media del 15% più lenta rispetto ai canali che non utilizzano conversazioni condivise.

Per assicurarsi che l'impostazione predefinita sia appropriata per una determinata istanza del canale, monitorare il modo in cui il canale viene eseguito con questa impostazione.

- Impostare un valore SHARECNV di 2 o superiore.

È possibile impostare SHARECNV (2) su SHARECNV (999999999). Per essere certi che l'impostazione scelta sia appropriata per una determinata istanza del canale, monitorare il modo in cui il canale viene eseguito con la nuova impostazione.

- Impostare un valore SHARECNV di 1.

Se non hai bisogno di conversazioni condivise, utilizza questa impostazione quando possibile. Elimina il conflitto per utilizzare il thread di ricezione e le applicazioni client possono sfruttare i miglioramenti delle prestazioni descritti nella sezione "Informazioni su questa attività".

Con questa impostazione, le prestazioni del server distribuito sono notevolmente migliorate. I miglioramenti delle prestazioni si applicano alle applicazioni client che emettono chiamate di attesa get sincrone non di lettura anticipata; ad esempio, le chiamate di attesa MQGET del client C. Quando queste applicazioni client sono connesse, il server distribuito utilizza meno thread e meno memoria e la velocità di trasmissione viene aumentata.

Se un server dispone di client connessi che condividono conversazioni su un socket e si diminuisce l'impostazione delle conversazioni condivise da SHARECNV (10) a SHARECNV (1), ciò ha i seguenti effetti:

- Aumento dell'utilizzo del socket sul server.
- Istanze di canale aumentate sul server.

In questo caso, è possibile anche scegliere di incrementare le impostazioni per **MaxChannels** e **MaxActiveChannels**.

Nota: È inoltre possibile impostare l'opzione MQCONN, MQCNO_NO_CONV_SHARING e collegare l'applicazione a un canale con **SHARECNV** impostato su un valore maggiore di 1. Il risultato è uguale alla connessione dell'applicazione a un canale con **SHARECNV** impostato su 1.

- Impostare un valore SHARECNV pari a 0.

L'istanza del canale si comporta esattamente come se fosse un canale di connessione client o server IBM WebSphere MQ 6.0. Non si ottengono conversazioni condivise o i miglioramenti delle prestazioni disponibili quando si imposta **SHARECNV** su 1 o su un valore superiore. Utilizzare il valore 0 solo se si dispone di applicazioni client esistenti che non vengono eseguite correttamente quando si imposta **SHARECNV** su 1 o su un valore superiore.

Concetti correlati

Client IBM MQ supportato: comportamento predefinito dei canali di connessione client e server

Ottimizzazione delle reti di pubblicazione / sottoscrizione distribuite

Utilizzare i suggerimenti di ottimizzazione in questa sezione per migliorare le prestazioni dei cluster e delle gerarchie di pubblicazione / sottoscrizione distribuiti IBM MQ .

Concetti correlati

[“Monitoraggio dei cluster” a pagina 319](#)

All'interno di un cluster è possibile monitorare i messaggi dell'applicazione, i messaggi di controllo e i log. Esistono considerazioni di monitoraggio speciali quando il cluster esegue il bilanciamento del carico tra due o più istanze di una coda.

Prestazioni cluster di pubblicazione / sottoscrizione instradate dirette

Nei cluster di pubblicazione / sottoscrizione instradati direttamente, le informazioni come gli argomenti del cluster e le sottoscrizioni proxy vengono inviate a tutti i membri del cluster, indipendentemente dal fatto che tutti i gestori code del cluster partecipino attivamente alla messaggistica di pubblicazione / sottoscrizione. Questo processo può creare un carico aggiuntivo significativo sul sistema. Per ridurre l'effetto della gestione del cluster sulle prestazioni, è possibile eseguire gli aggiornamenti in orari non di picco, definire un sottoinsieme molto più piccolo di gestori code coinvolti nella pubblicazione / sottoscrizione e renderlo un cluster "sovrapposto" oppure passare all'utilizzo dell'instradamento dell'host argomento.

Esistono due origini del carico di lavoro su un gestore code in un cluster di pubblicazione / sottoscrizione:

- Gestione diretta dei messaggi per i programmi applicativi.
- Gestione di messaggi e canali necessari per gestire il cluster.

In un tipico cluster point - to - point, il carico di lavoro del sistema cluster è in gran parte limitato alle informazioni esplicitamente richieste dai membri del cluster come richiesto. Pertanto, in qualsiasi cosa diversa da un cluster point-to-point molto grande, ad esempio uno che contiene migliaia di gestori code, è possibile in gran parte non considerare l'effetto delle prestazioni della gestione del cluster. Tuttavia, in un cluster di pubblicazione / sottoscrizione instradato direttamente, le informazioni quali gli argomenti del cluster, l'appartenenza al gestore code e le sottoscrizioni proxy vengono inviate a tutti i membri del cluster, indipendentemente dal fatto che tutti i gestori code del cluster partecipino attivamente alla messaggistica di pubblicazione / sottoscrizione. Ciò può creare un carico aggiuntivo significativo sul sistema. Pertanto, è necessario considerare l'effetto della gestione del cluster sulle prestazioni del gestore code, sia nella tempistica che nella dimensione.

Caratteristiche delle prestazioni dei cluster instradati direttamente

Confrontare un cluster point-to-point con un cluster di pubblicazione / sottoscrizione instradato direttamente rispetto alle attività di gestione principali.

Innanzitutto, un cluster punto a punto:

1. Quando viene definita una nuova coda cluster, le informazioni di destinazione vengono inoltrate ai gestori code del repository completo e inviate solo ad altri membri del cluster quando fanno riferimento per la prima volta a una coda cluster (ad esempio, quando un'applicazione tenta di aprirla). Queste informazioni vengono quindi memorizzate nella cache in locale dal gestore code per eliminare la necessità di richiamare in remoto le informazioni ogni volta che si accede alla coda.
2. L'aggiunta di un gestore code ad un cluster non influisce direttamente sul carico su altri gestori code. Le informazioni sul nuovo gestore code vengono inoltrate ai repository completi, ma i canali al nuovo gestore code da altri gestori code nel cluster vengono creati e avviati solo quando il traffico inizia a fluire verso o dal nuovo gestore code.

In sintesi, il carico su un gestore code in un cluster point - to - point è correlato al traffico messaggi che gestisce per i programmi applicativi e non è direttamente correlato alle dimensioni del cluster.

Secondo, un cluster di pubblicazione / sottoscrizione instradato direttamente:

1. Quando viene definito un nuovo argomento del cluster, le informazioni vengono inoltrate ai gestori code del repository completo e da lì direttamente a tutti i membri del cluster, causando l'avvio dei

canali per ciascun membro del cluster dai repository completi, se non già avviati. Se questo è il primo argomento cluster diretto, a ciascun membro del gestore code vengono inviate informazioni su tutti gli altri membri del gestore code nel cluster.

2. Quando una sottoscrizione viene creata a un argomento cluster su una nuova stringa di argomenti, le informazioni vengono inviate direttamente da tale gestore code a tutti gli altri membri del cluster immediatamente, causando l'avvio dei canali per ciascun membro del cluster da tale gestore code, se non è già stato avviato.
3. Quando un nuovo gestore code si unisce a un cluster esistente, le informazioni su tutti gli argomenti del cluster (e su tutti i membri del gestore code se è definito un argomento del cluster diretto) vengono inviate al nuovo gestore code dai gestori code del repository completo. Il nuovo gestore code sincronizza la conoscenza di tutte le sottoscrizioni agli argomenti del cluster nel cluster con tutti i membri del cluster.

In sintesi, il carico di gestione del cluster su qualsiasi gestore code in un cluster di pubblicazione / sottoscrizione instradato direttamente aumenta con il numero di gestori code, gli argomenti raggruppati in cluster e le modifiche alle sottoscrizioni su diverse stringhe di argomenti all'interno del cluster, indipendentemente dall'utilizzo locale di tali argomenti su ciascun gestore code.

In un cluster di grandi dimensioni, o in uno in cui la frequenza di modifica delle sottoscrizioni è elevata, questo livello di gestione del cluster può essere un sovraccarico significativo su tutti i gestori code.

Riduzione dell'effetto della pubblicazione / sottoscrizione diretta instradata sulle prestazioni

Per ridurre l'effetto della gestione del cluster sulle prestazioni di un cluster di pubblicazione / sottoscrizione instradato direttamente, considerare le seguenti opzioni:

- Eseguire gli aggiornamenti di cluster, argomenti e sottoscrizioni in orari non di picco del giorno.
- Definire un sottoinsieme molto più piccolo di gestori code coinvolti nella pubblicazione / sottoscrizione e creare un cluster "sovrapposto". Questo cluster è quindi il cluster in cui sono definiti gli argomenti del cluster. Sebbene alcuni gestori code si trovino ora in due cluster, l'effetto complessivo della pubblicazione / sottoscrizione è ridotto:
 - La dimensione del cluster di pubblicazione / sottoscrizione è inferiore.
 - I gestori code che non si trovano nel cluster di pubblicazione / sottoscrizione sono molto meno interessati dal traffico di gestione del cluster.

Se le opzioni precedenti non risolvono adeguatamente i tuoi problemi di prestazioni, prendi in considerazione l'utilizzo di un cluster di pubblicazione / sottoscrizione *host argomento instradato*. Per un confronto dettagliato dell'instradamento diretto e dell'instradamento host argomento nei cluster di pubblicazione / sottoscrizione, consultare [Progettazione di cluster di pubblicazione / sottoscrizione](#).

Concetti correlati

Prestazioni del cluster di pubblicazione / sottoscrizione instradato dell'host argomento

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento fornisce un controllo preciso sui gestori code che ospitano ciascun argomento. Questi host argomento diventano i gestori code di *instradamento* per quel ramo della struttura argomento. Inoltre, i gestori code senza sottoscrizioni o publisher non hanno bisogno di connettersi con gli host argomento. Questa configurazione può ridurre significativamente il numero di connessioni tra i gestori code nel cluster e la quantità di informazioni trasmesse tra i gestori code.

Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione

Un concetto importante nelle prestazioni di messaggistica asincrona è *balance*. A meno che i destinatari dei messaggi non siano bilanciati con i produttori di messaggi, esiste il rischio che un backlog di messaggi non utilizzati possa accumularsi e compromettere seriamente le prestazioni di più applicazioni.

Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione

La pubblicazione / sottoscrizione distribuita in IBM MQ funziona propagando la conoscenza del punto in cui le sottoscrizioni a diverse stringhe di argomenti sono state create nella rete del gestore code. Ciò

consente al gestore code su cui viene pubblicato un messaggio di identificare quali altri gestori code richiedono una copia del messaggio pubblicato, in modo che corrispondano alle relative sottoscrizioni.

Prestazioni del cluster di pubblicazione / sottoscrizione instradato dell'host argomento

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento fornisce un controllo preciso sui gestori code che ospitano ciascun argomento. Questi host argomento diventano i gestori code di *instradamento* per quel ramo della struttura argomento. Inoltre, i gestori code senza sottoscrizioni o publisher non hanno bisogno di connettersi con gli host argomento. Questa configurazione può ridurre significativamente il numero di connessioni tra i gestori code nel cluster e la quantità di informazioni trasmesse tra i gestori code.

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento si comporta come segue:

- Gli argomenti vengono definiti manualmente sui singoli *gestori code dell'host argomento* nel cluster.
- Quando viene effettuata una sottoscrizione su un gestore code del cluster, le sottoscrizioni proxy vengono create solo sugli host argomento.
- Quando un'applicazione pubblica le informazioni su un argomento, il gestore code di ricezione inoltra la pubblicazione a un gestore code che ospita l'argomento. L'host argomento invia quindi la pubblicazione a tutti i gestori code nel cluster che hanno sottoscrizioni valide all'argomento.

Per un'introduzione più dettagliata all'instradamento dell'host argomento, consultare [Instradamento dell'host argomento nei cluster](#).

Per molte configurazioni, l'instradamento host argomento è una topologia più appropriata rispetto all'*instradamento diretto* poiché fornisce i seguenti vantaggi:

- Migliore scalabilità dei cluster di grandi dimensioni. Solo i gestori code dell'host argomento devono essere in grado di connettersi a tutti gli altri gestori code nel cluster. Pertanto, ci sono meno canali tra i gestori code e un minore traffico amministrativo di pubblicazione/sottoscrizione che per l'instradamento diretto. Quando le sottoscrizioni subiscono una modifica su un gestore code, devono essere informati solo i gestori code dell'host argomento.
- Più controllo sulla configurazione fisica. Con l'instradamento diretto, ogni gestore code assume tutti i ruoli e quindi tutti i gestori devono essere ugualmente capaci. Con l'instradamento all'host argomento, si scelgono in modo esplicito i gestori code dell'host argomento. Pertanto, è possibile garantire che tali gestori code siano in esecuzione sul dispositivo adeguato ed è possibile utilizzare sistemi meno potenti per gli altri gestori code.

Tuttavia, l'instradamento all'host argomento impone anche alcuni vincoli sul sistema:

- La configurazione e manutenzione del sistema richiedono una maggiore pianificazione rispetto all'instradamento diretto. È necessario decidere quali punti raggruppare nella struttura ad albero degli argomenti e la posizione delle definizioni di argomento nel cluster.
- Così come per gli argomenti con instradamento diretto, quando si definisce un nuovo argomento instradato all'host argomento, le informazioni vengono inviate ai gestori code del repository completo e da lì indirizzate a tutti i membri del cluster. Questo evento fa sì che i canali vengano avviati in ciascun membro del cluster da tutti i repository (se non già avviati).
- Le pubblicazioni vengono sempre inviate al gestore code host da un gestore code non host, anche se non sono presenti sottoscrizioni nel cluster. Pertanto, occorre utilizzare gli argomenti instradati quando è prevista l'esistenza di sottoscrizioni o quando il sovraccarico di connettività globale e conoscenza è maggiore del rischio di traffico di pubblicazione supplementare.
- I messaggi che vengono pubblicati sui gestori code non host non vengono indirizzati direttamente al gestore code che ospita la sottoscrizione, ma vengono instradati sempre attraverso un gestore code dell'host argomento. Questo approccio può aumentare il sovraccarico totale nel cluster, nonché aumentare la latenza dei messaggi e ridurre le prestazioni.

Nota: Per alcune configurazioni, è possibile rimuovere utilmente questo vincolo come descritto in [Instradamento host argomento utilizzando publisher o sottoscrittori centralizzati](#).

- L'utilizzo di un unico gestore code dell'host argomento introduce un singolo punto di errore per tutti i messaggi che vengono pubblicati in un argomento. È possibile rimuovere questo singolo punto di errore definendo più host argomento. Tuttavia, la presenza di più host influisce sull'ordine dei messaggi pubblicati man mano che vengono ricevuti dalle sottoscrizioni.
- Il carico di messaggi supplementare è sostenuto dai gestori code dell'host argomento, in quanto questi dovranno elaborare il traffico di pubblicazione da più gestori code. Questo carico può essere ridotto: utilizzare più host argomento per un singolo argomento (nel qual caso l'ordine dei messaggi non viene mantenuto) o utilizzare gestori code differenti per ospitare gli argomenti instradati per i diversi rami di una struttura ad albero degli argomenti.

Instradamento host argomento con publisher o sottoscrittori centralizzati

Per rimuovere l'"hop" supplementare che si verifica quando le pubblicazioni vengono sempre instradate alle sottoscrizioni tramite un gestore code dell'host argomento, configurare i publisher o le sottoscrizioni sullo stesso gestore code che ospita l'argomento. Questo approccio offre i massimi vantaggi in termini di prestazioni nei seguenti due casi:

- Argomenti con molti publisher e poche sottoscrizioni. In questo caso, ospitare le sottoscrizioni sul gestore code dell'host argomento.
- Argomenti con pochi publisher e molte sottoscrizioni. In questo caso, ospitare i publisher sul gestore code dell'host argomento.

La seguente figura mostra un gestore code dell'host argomento che ospita anche le sottoscrizioni. Questo approccio rimuove l'ulteriore "hop" tra il publisher e il sottoscrittore e riduce la condivisione non necessaria della conoscenza della sottoscrizione tra tutti i membri del cluster:

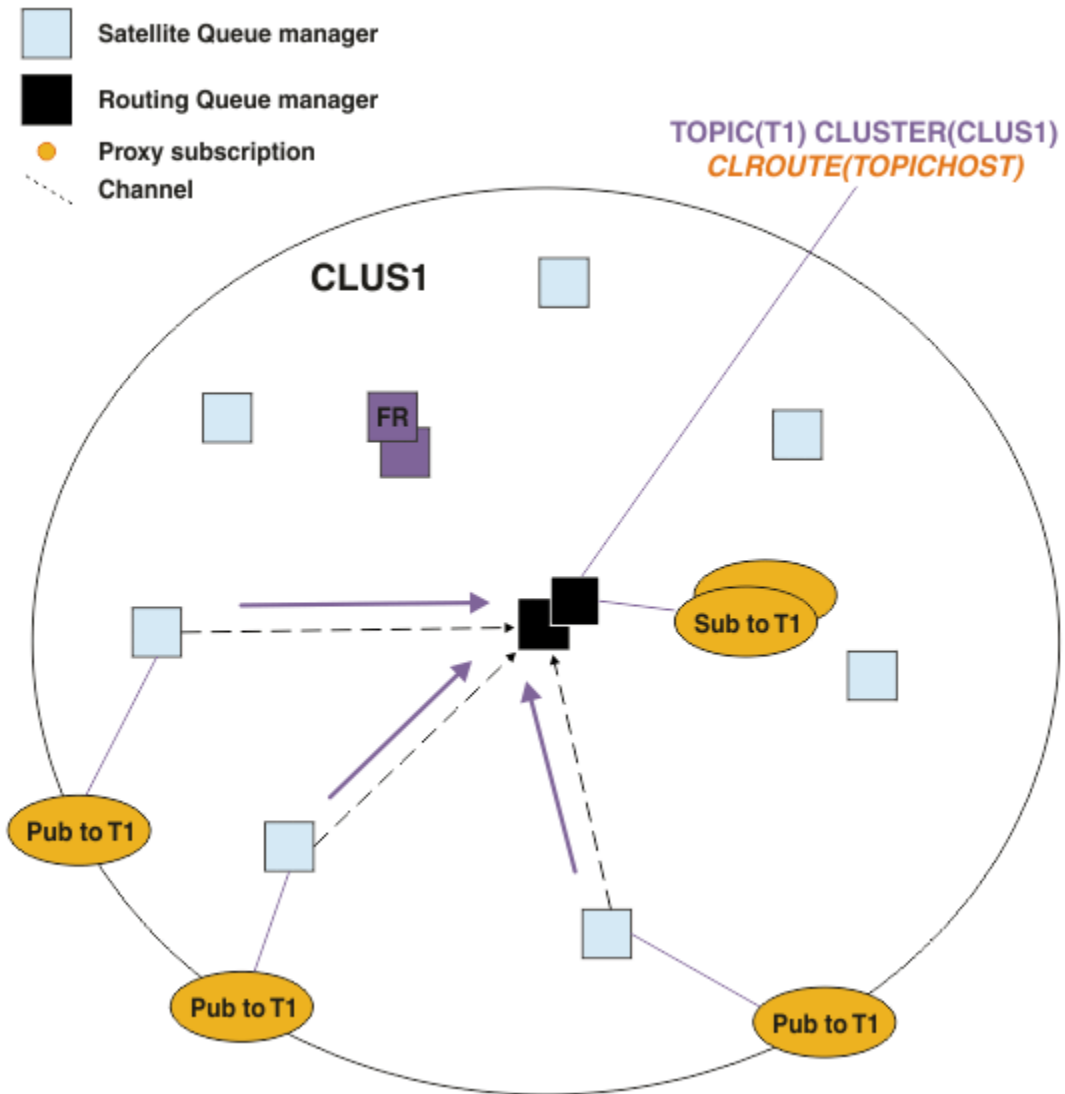


Figura 27. Hosting delle sottoscrizioni su un gestore code dell'host argomento

La seguente figura mostra un gestore code dell'host argomento che ospita anche i publisher. Questo approccio rimuove l'ulteriore "hop" tra il publisher e il sottoscrittore e riduce la condivisione non necessaria della conoscenza della sottoscrizione tra tutti i membri del cluster:

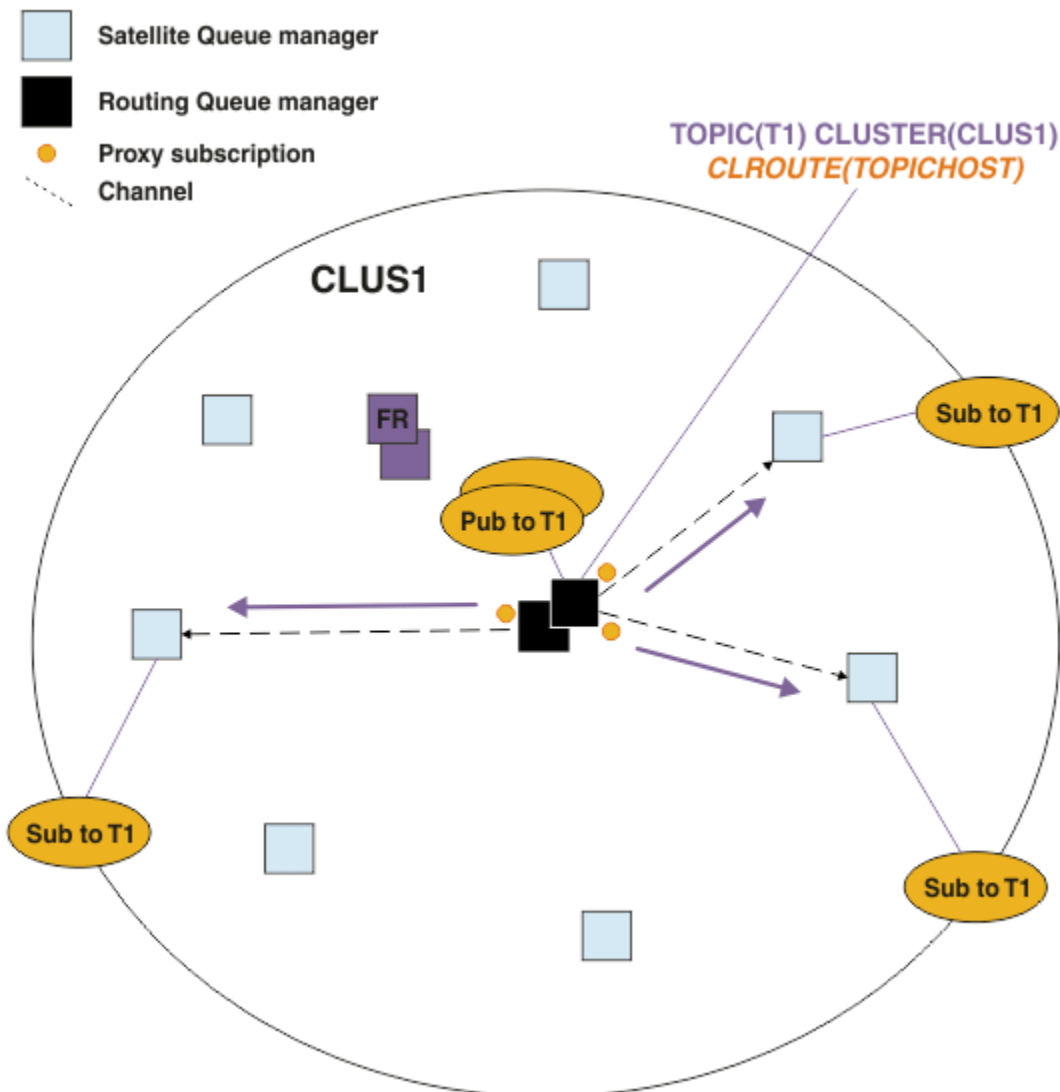


Figura 28. Hosting delle pubblicazioni su un gestore code dell'host argomento

Concetti correlati

Prestazioni cluster di pubblicazione / sottoscrizione instradate dirette

Nei cluster di pubblicazione / sottoscrizione instradati direttamente, le informazioni come gli argomenti del cluster e le sottoscrizioni proxy vengono inviate a tutti i membri del cluster, indipendentemente dal fatto che tutti i gestori code del cluster partecipino attivamente alla messaggistica di pubblicazione / sottoscrizione. Questo processo può creare un carico aggiuntivo significativo sul sistema. Per ridurre l'effetto della gestione del cluster sulle prestazioni, è possibile eseguire gli aggiornamenti in orari non di picco, definire un sottoinsieme molto più piccolo di gestori code coinvolti nella pubblicazione / sottoscrizione e renderlo un cluster "sovrapposto" oppure passare all'utilizzo dell'instradamento dell'host argomento.

Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione

Un concetto importante nelle prestazioni di messaggistica asincrona è *balance*. A meno che i destinatari dei messaggi non siano bilanciati con i produttori di messaggi, esiste il rischio che un backlog di messaggi non utilizzati possa accumularsi e compromettere seriamente le prestazioni di più applicazioni.

Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione

La pubblicazione / sottoscrizione distribuita in IBM MQ funziona propagando la conoscenza del punto in cui le sottoscrizioni a diverse stringhe di argomenti sono state create nella rete del gestore code. Ciò consente al gestore code su cui viene pubblicato un messaggio di identificare quali altri gestori code richiedono una copia del messaggio pubblicato, in modo che corrispondano alle relative sottoscrizioni.

Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione

Un concetto importante nelle prestazioni di messaggistica asincrona è *balance*. A meno che i destinatari dei messaggi non siano bilanciati con i produttori di messaggi, esiste il rischio che un backlog di messaggi non utilizzati possa accumularsi e compromettere seriamente le prestazioni di più applicazioni.

In una topologia di messaggistica point - to - point, la relazione tra i destinatari e i produttori di messaggi è facilmente comprensibile. È possibile ottenere stime di produzione e consumo dei messaggi, coda per coda, canale per canale. Se vi è una mancanza di equilibrio, le strozzature vengono prontamente individuate e quindi risolte.

È più difficile stabilire se i publisher e i sottoscrittori sono bilanciati in una topologia di pubblicazione / sottoscrizione. Iniziare da ogni sottoscrizione e tornare ai gestori code con i publisher sull'argomento. Calcolare il numero di pubblicazioni in transito per ciascun sottoscrittore da ciascun gestore code.

Ogni pubblicazione che corrisponde a una sottoscrizione su un gestore code remoto (basata su sottoscrizioni proxy) viene inserita in una coda di trasmissione. Se più gestori code remoti dispongono di sottoscrizioni proxy per tale pubblicazione, più copie del messaggio vengono inserite in una coda di trasmissione, ciascuna destinata a un canale mittente differente.

In un cluster di pubblicazione / sottoscrizione, tali pubblicazioni sono destinate alla coda SYSTEM . INTER . QMGR . PUBS sui gestori code remoti che ospitano le sottoscrizioni. In una gerarchia, ogni pubblicazione è destinata alla coda SYSTEM . BROKER . DEFAULT . STREAM o a qualsiasi altra coda di flusso elencata in SYSTEM . QPUBSUB . QUEUE . NAMELIST sui gestori code remoti. Ciascun gestore code elabora i messaggi in arrivo su tale coda e li consegna alle sottoscrizioni corrette su tale gestore code.

Per questo motivo, monitorare il carico nei seguenti punti in cui potrebbero verificarsi colli di bottiglia:

- Monitorare il carico nelle singole code di sottoscrizione.
 - Questo collo di bottiglia implica che l'applicazione di sottoscrizione non utilizza le pubblicazioni con la stessa rapidità con cui vengono pubblicate.
- Monitorare il carico nella coda SYSTEM . INTER . QMGR . PUBS o nelle code di flusso.
 - Questo collo di bottiglia implica che il gestore code riceve le pubblicazioni da uno o più gestori code remoti più velocemente di quanto possa distribuirle alle sottoscrizioni locali.
 - Quando viene visualizzato su un gestore code dell'host argomento quando si utilizza l'instradamento dell'host argomento in un cluster, considerare la possibilità di creare ulteriori host argomento dei gestori code, consentendo il bilanciamento del carico di lavoro di pubblicazione. Tuttavia, ciò influenzerà l'ordine dei messaggi tra le pubblicazioni. Consultare [Instradamento host argomento utilizzando più host argomento per un singolo argomento](#).
- Monitorare il carico sui canali tra il gestore code di pubblicazione e i gestori code di sottoscrizione, alimentati dalle code di trasmissione sul gestore code di pubblicazione.
 - Questo collo di bottiglia implica che uno o più canali non sono in esecuzione o che i messaggi vengono pubblicati sul gestore code locale più velocemente di quanto i canali possano consegnare al gestore code remoto.
 - Quando si utilizza un cluster di pubblicazione / sottoscrizione, definire ulteriori canali riceventi del cluster sul gestore code di destinazione. Ciò consente il bilanciamento del carico di lavoro della pubblicazione. Tuttavia, ciò influenza l'ordinamento dei messaggi tra le pubblicazioni. Considerare inoltre di passare a una configurazione di più code di trasmissione del cluster, in quanto ciò può migliorare le prestazioni in determinate circostanze.
- Se l'applicazione di pubblicazione sta utilizzando un'interfaccia di pubblicazione / sottoscrizione accodata, monitorare il carico alla (a) coda SYSTEM . BROKER . DEFAULT . STREAM e qualsiasi altra coda di flusso elencata in SYSTEM . QPUBSUB . QUEUE . NAMELIST ; e (b) la coda SYSTEM . BROKER . DEFAULT . SUBPOINT e tutte le altre code di punti secondari elencate in SYSTEM . QPUBSUB . SUBPOINT . NAMELIST .
 - Questo collo di bottiglia implica che i messaggi vengono inseriti dalle applicazioni di pubblicazione locali più velocemente di quanto il gestore code locale possa elaborare i messaggi.

Concetti correlati

Prestazioni cluster di pubblicazione / sottoscrizione instradate dirette

Nei cluster di pubblicazione / sottoscrizione instradati direttamente, le informazioni come gli argomenti del cluster e le sottoscrizioni proxy vengono inviate a tutti i membri del cluster, indipendentemente dal fatto che tutti i gestori code del cluster partecipino attivamente alla messaggistica di pubblicazione / sottoscrizione. Questo processo può creare un carico aggiuntivo significativo sul sistema. Per ridurre l'effetto della gestione del cluster sulle prestazioni, è possibile eseguire gli aggiornamenti in orari non di picco, definire un sottoinsieme molto più piccolo di gestori code coinvolti nella pubblicazione / sottoscrizione e renderlo un cluster "sovrapposto" oppure passare all'utilizzo dell'instradamento dell'host argomento.

Prestazioni del cluster di pubblicazione / sottoscrizione instradato dell'host argomento

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento fornisce un controllo preciso sui gestori code che ospitano ciascun argomento. Questi host argomento diventano i gestori code di *instradamento* per quel ramo della struttura argomento. Inoltre, i gestori code senza sottoscrizioni o publisher non hanno bisogno di connettersi con gli host argomento. Questa configurazione può ridurre significativamente il numero di connessioni tra i gestori code nel cluster e la quantità di informazioni trasmesse tra i gestori code.

Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione

La pubblicazione / sottoscrizione distribuita in IBM MQ funziona propagando la conoscenza del punto in cui le sottoscrizioni a diverse stringhe di argomenti sono state create nella rete del gestore code. Ciò consente al gestore code su cui viene pubblicato un messaggio di identificare quali altri gestori code richiedono una copia del messaggio pubblicato, in modo che corrispondano alle relative sottoscrizioni.

“Monitoraggio dei cluster” a pagina 319

All'interno di un cluster è possibile monitorare i messaggi dell'applicazione, i messaggi di controllo e i log. Esistono considerazioni di monitoraggio speciali quando il cluster esegue il bilanciamento del carico tra due o più istanze di una coda.

Prestazioni della sottoscrizione nelle reti di pubblicazione / sottoscrizione

La pubblicazione / sottoscrizione distribuita in IBM MQ funziona propagando la conoscenza del punto in cui le sottoscrizioni a diverse stringhe di argomenti sono state create nella rete del gestore code. Ciò consente al gestore code su cui viene pubblicato un messaggio di identificare quali altri gestori code richiedono una copia del messaggio pubblicato, in modo che corrispondano alle relative sottoscrizioni.

Questo approccio riduce al minimo l'invio di messaggi pubblicati ai gestori code su cui non esistono sottoscrizioni corrispondenti. Tuttavia, la propagazione della conoscenza della sottoscrizione può diventare un sovraccarico significativo, quando il numero di stringhe di argomenti sottoscritte è elevato e cambia costantemente attraverso la creazione e l'eliminazione di sottoscrizioni frequenti.

È possibile influenzare le prestazioni modificando il modo in cui le pubblicazioni e le sottoscrizioni vengono spostate sulla rete di pubblicazione / sottoscrizione. Se il traffico di rete dispone di poche pubblicazioni e di una rapida creazione, eliminazione o modifica delle sottoscrizioni, è possibile arrestare il flusso delle informazioni sulle sottoscrizioni a tutti i gestori code e, invece, inoltrare tutte le pubblicazioni a tutti i gestori code nella rete. È inoltre possibile limitare il flusso di sottoscrizioni proxy e pubblicazioni per un determinato argomento tra gestori code connessi, limitare il flusso di sottoscrizioni proxy contenenti caratteri jolly e ridurre il numero e la natura transitoria delle stringhe argomento.

Propagazione della sottoscrizione individuale e *pubblicazione ovunque*

Pubblica ovunque è un'alternativa alla singola propagazione della sottoscrizione. Con la propagazione individuale, solo le pubblicazioni che hanno una sottoscrizione corrispondente su un gestore code vengono inoltrate a tale gestore code. Con *pubblica ovunque*, tutte le pubblicazioni vengono inoltrate a tutti i gestori code nella rete. I gestori code riceventi consegnano quindi le pubblicazioni che corrispondono alle sottoscrizioni locali.

Propagazione di singole sottoscrizioni

Questo meccanismo determina la minore quantità di traffico di pubblicazioni tra gestori code, poiché vengono inviate solo le pubblicazioni che corrispondono alle sottoscrizioni su un gestore code.

Tuttavia:

- Per ogni singola stringa argomento sottoscritta, una sottoscrizione proxy viene inviata ad altri gestori code nella topologia di pubblicazione / sottoscrizione. La serie di gestori code dipende dal modello di instradamento utilizzato, come descritto in [Pianificazione della rete di pubblicazione / sottoscrizione distribuita](#).
 - Questo sovraccarico di messaggistica può essere significativo se ci sono molte migliaia di sottoscrizioni da creare o eliminare (ad esempio, ricreando tutte le sottoscrizioni non durevoli dopo un riavvio di un gestore code) o se la serie di sottoscrizioni sta cambiando rapidamente e ciascuna corrisponde a una stringa di argomenti diversa.
 - Il numero di gestori code a cui viene propagata la sottoscrizione proxy influenza anche la scala del sovraccarico.
- Le sottoscrizioni proxy vengono fatte passare ad altri gestori code utilizzando la messaggistica asincrona. Ciò ha il seguente effetto:
 - Esiste un ritardo tra la creazione di una sottoscrizione e la creazione, la consegna e l'elaborazione della sottoscrizione proxy da parte degli altri gestori code.
 - I messaggi pubblicati sui gestori code in tale intervallo non vengono consegnati alla sottoscrizione remota.

Pubblica ovunque

Con questo meccanismo non esiste alcun sovraccarico di sottoscrizione proxy per stringa di argomenti sul sistema. Ciò significa che la rapida creazione, eliminazione o modifica della sottoscrizione non comporta un aumento del carico di rete e dell'elaborazione.

Inoltre, non vi è alcun ritardo tra la creazione di una sottoscrizione e il flusso delle pubblicazioni a un gestore code, poiché tutte le pubblicazioni vengono trasferiti a tutti i gestori code. Pertanto, non esiste alcuna finestra in cui le pubblicazioni non vengano consegnate alle sottoscrizioni remote appena create.

Tuttavia:

- L'invio di tutte le pubblicazioni a tutti i gestori code nella topologia di pubblicazione / sottoscrizione può causare un traffico di rete eccessivo in cui le pubblicazioni non hanno sottoscrizioni corrispondenti su ciascun gestore code.
 - Maggiore è il numero di gestori code nella topologia, maggiore è il sovraccarico.

Si consiglia di utilizzare il meccanismo *pubblica ovunque* quando si prevede che una pubblicazione venga sottoscritta da una percentuale significativa dei gestori code o quando i costi generali della sottoscrizione proxy sono troppo grandi a causa della frequenza delle modifiche della sottoscrizione. È necessario utilizzare l'inoltro di singole sottoscrizioni proxy nei casi in cui si riscontra un aumento del traffico di messaggistica quando le pubblicazioni vengono inviate a tutti i gestori code, piuttosto che ai gestori code con sottoscrizioni corrispondenti.

È possibile impostare il comportamento *pubblica ovunque* a qualsiasi livello all'interno della struttura ad albero degli argomenti. Per abilitare la *pubblicazione ovunque*, impostare il parametro **PROXYSUB** su **FORCE** per un oggetto argomento di alto livello. Ne risulta una singola sottoscrizione proxy con caratteri jolly che corrisponde a tutti gli argomenti al di sotto di questo oggetto argomento nella struttura ad albero degli argomenti. Quando viene impostato su un oggetto argomento in cluster, l'attributo **PROXYSUB (FORCE)** viene propagato a tutti i gestori code della rete, non solo al gestore code su cui è stato definito l'argomento.

Nota: Quando viene utilizzato in una gerarchia, si imposta **PROXYSUB (FORCE)** singolarmente su ciascun gestore code, in modo che il meccanismo della topologia limiti naturalmente il numero di canali. Tuttavia, quando viene utilizzato in un cluster, potrebbero essere avviati molti canali aggiuntivi:

- In un cluster instradato all'host argomento, i canali vengono avviati da ciascun gestore code a ciascun gestore code dell'host argomento.
- In un cluster instradato direttamente, i canali vengono avviati da ogni gestore code a ogni altro gestore code.

Il sovraccarico dell'avvio di molti canali è più pronunciato in un cluster instradato direttamente e può causare problemi di prestazioni. Consultare [“Prestazioni cluster di pubblicazione / sottoscrizione instradate dirette”](#) a pagina 383.

Altri modi per limitare il flusso di sottoscrizioni proxy e pubblicazioni tra gestori code connessi

Consolida stringhe argomento

L'utilizzo di molte stringhe di argomenti distinte, transitorie, introduce un certo livello di overhead di gestione su ciascun gestore code nel sistema in cui sono collegati i publisher o le sottoscrizioni. È necessario valutare periodicamente l'utilizzo delle stringhe di argomenti per verificare se possono essere consolidate. La riduzione del numero e della natura transitoria delle stringhe di argomenti, e quindi dei publisher e delle relative sottoscrizioni, riduce l'impatto sul sistema.

Limita ambito di pubblicazione e sottoscrizione

Per un determinato argomento, è possibile utilizzare le impostazioni [Ambito pubblicazione](#) e [Ambito sottoscrizione](#) per mantenere le pubblicazioni e le sottoscrizioni locali per il gestore code su cui sono definite.

Blocca sottoscrizioni effettuate ad argomenti con caratteri jolly

È possibile limitare il flusso delle sottoscrizioni proxy contenenti caratteri jolly impostando l'attributo **Topic** WILDCARD su BLOCK. Consultare [Carattere jolly nelle sottoscrizioni proxy](#).

Vedere anche [“Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione”](#) a pagina 389

Monitoraggio del traffico di sottoscrizione proxy nei cluster

Quando si considera il carico sul sistema dal traffico di sottoscrizione proxy, oltre al monitoraggio delle code elencate nella [“Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione”](#) a pagina 389, monitorare le seguenti code cluster:

- Il SISTEMA SYSTEM.INTER.QMGR.FANREQ sul gestore code del sottoscrittore.
- Il SISTEMA SYSTEM.INTER.QMGR.CONTROL su tutti gli altri gestori code nel cluster.

Qualsiasi backlog di messaggi significativi su queste code implica che la frequenza di modifica della sottoscrizione è troppo elevata per il sistema o che un gestore code non funziona correttamente nel cluster. Se si sospetta che il problema risieda in un gestore code specifico, verificare che il supporto di pubblicazione / sottoscrizione non sia disabilitato per tale gestore code. Consultare **PSMODE** in [ALTER QMGR](#).

Concetti correlati

[Prestazioni cluster di pubblicazione / sottoscrizione instradate dirette](#)

Nei cluster di pubblicazione / sottoscrizione instradati direttamente, le informazioni come gli argomenti del cluster e le sottoscrizioni proxy vengono inviate a tutti i membri del cluster, indipendentemente dal fatto che tutti i gestori code del cluster partecipino attivamente alla messaggistica di pubblicazione / sottoscrizione. Questo processo può creare un carico aggiuntivo significativo sul sistema. Per ridurre l'effetto della gestione del cluster sulle prestazioni, è possibile eseguire gli aggiornamenti in orari non di picco, definire un sottoinsieme molto più piccolo di gestori code coinvolti nella pubblicazione / sottoscrizione e renderlo un cluster "sovrapposto" oppure passare all'utilizzo dell'instradamento dell'host argomento.

[Prestazioni del cluster di pubblicazione / sottoscrizione instradato dell'host argomento](#)

Un cluster di pubblicazione / sottoscrizione instradato dell'host argomento fornisce un controllo preciso sui gestori code che ospitano ciascun argomento. Questi host argomento diventano i gestori code di *instradamento* per quel ramo della struttura argomento. Inoltre, i gestori code senza sottoscrizioni o publisher non hanno bisogno di connettersi con gli host argomento. Questa configurazione può ridurre significativamente il numero di connessioni tra i gestori code nel cluster e la quantità di informazioni trasmesse tra i gestori code.

[Equilibrare produttori e consumatori nelle reti di pubblicazione / sottoscrizione](#)

Un concetto importante nelle prestazioni di messaggistica asincrona è *balance*. A meno che i destinatari dei messaggi non siano bilanciati con i produttori di messaggi, esiste il rischio che un backlog di messaggi non utilizzati possa accumularsi e compromettere seriamente le prestazioni di più applicazioni.

[Sottoscrizioni proxy in una rete di pubblicazione / sottoscrizione](#)

Riduzione del numero di argomenti indesiderati nella struttura ad albero degli argomenti

Le prestazioni di un sistema di pubblicazione / sottoscrizione sono migliorate riducendo il numero di argomenti indesiderati nella struttura ad albero degli argomenti. Che cosa è un argomento indesiderato e come si rimuovono?

È possibile creare un numero elevato di argomenti senza influire negativamente sulle prestazioni. Tuttavia, alcuni modi di utilizzare la pubblicazione / sottoscrizione comportano un'espansione continua delle strutture ad albero degli argomenti. Un numero eccezionalmente elevato di argomenti viene creato una volta e non viene mai più utilizzato. Il numero crescente di argomenti potrebbe diventare un problema di prestazioni.

Come si può evitare i disegni che portano a un numero crescente di argomenti indesiderati? Cosa è possibile fare per aiutare il gestore code a rimuovere gli argomenti indesiderati dalla struttura ad albero degli argomenti?

Il gestore code riconosce un argomento indesiderato perché è stato inutilizzato per 30 minuti. Il gestore code rimuove per conto dell'utente gli argomenti non utilizzati dalla struttura ad albero degli argomenti. La durata di 30 minuti può essere modificata modificando l'attributo del gestore code, **TREELIFE**. È possibile aiutare il gestore code a rimuovere gli argomenti indesiderati accertandosi che l'argomento appaia inutilizzato al gestore code. La sezione, ["Cos' è un argomento inutilizzato?"](#) a pagina 393 spiega cos' è un argomento non utilizzato.

Un programmatore, progettando qualsiasi applicazione, e soprattutto progettando un'applicazione di lunga durata, considera il suo utilizzo di risorse: quanta risorsa richiede il programma, ci sono richieste illimitate e perdite di risorse? Gli argomenti sono una risorsa utilizzata dai programmi di pubblicazione / sottoscrizione. Esamina l'uso degli argomenti come qualsiasi altra risorsa che un programma utilizza.

Cos' è un argomento inutilizzato?

Prima di definire cos' è un argomento inutilizzato, cosa conta esattamente come argomento?

Quando una stringa di argomenti, come USA/Alabama/Auburn, viene convertita in un argomento, l'argomento viene aggiunto alla struttura ad albero degli argomenti. Se necessario, nella struttura ad albero vengono creati ulteriori nodi di argomenti e gli argomenti corrispondenti. La stringa di argomenti USA/Alabama/Auburn viene convertita in una struttura ad albero con tre argomenti.

- USA
- USA/Alabama
- USA/Alabama/Auburn

Per visualizzare tutti gli argomenti nella struttura ad albero degli argomenti, utilizzare il comando **runmqsc DISPLAY TPSTATUS(' # ') TYPE(TOPIC)**.

Un argomento non utilizzato nella struttura ad albero degli argomenti ha le seguenti proprietà.

Non è associato a un oggetto argomento

Un oggetto argomento di gestione ha una stringa di argomento che lo associa a un argomento. Quando si definisce l'oggetto argomento Alabama, se l'argomento, USA/Alabama, a cui deve essere associato, non esiste, l'argomento viene creato dalla stringa di argomenti. Se l'argomento esiste, l'oggetto argomento e l'argomento vengono associati insieme utilizzando la stringa dell'argomento.

Non dispone di una pubblicazione conservata

Un argomento con una pubblicazione conservata risulta da un publisher che immette un messaggio in un argomento con opzione MQPMO_RETAIN.

Utilizzare il **runmqsc** comando `DISPLAY TPSTATUS('USA/Alabama') RETAINED` per verificare se USA/Alabama dispone di una pubblicazione conservata. La risposta è YES o NO.

Utilizzare il **runmqsc** comando `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)` per rimuovere una pubblicazione conservata da USA/Alabama.

Non ha argomenti secondari

USA/Alabama/Auburn è un argomento senza argomenti secondari. USA/Alabama/Auburn è l'argomento secondario diretto di USA/Alabama.

Visualizzare gli elementi child diretti di USA/Alabama con il comando **runmqsc** `DISPLAY TPSTATUS('USA/Alabama/+')`.

Nessun publisher attivo sul nodo

Un publisher attivo su un nodo è un'applicazione che ha l'argomento aperto per l'output.

Ad esempio, un'applicazione apre l'oggetto argomento denominato **Alabama** con opzioni di apertura MQOO_OUTPUT.

Per visualizzare i publisher attivi in USA/Alabama e in tutti i relativi child, utilizzare il comando **runmqsc** `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN`.

Nessun sottoscrittore attivo al nodo

Un sottoscrittore attivo può essere una sottoscrizione duratura o un'applicazione che ha registrato una sottoscrizione a un argomento con MQSUBe non l'ha chiusa.

Per visualizzare le sottoscrizioni attive in USA/Alabama, utilizzare il **runmqsc** comando `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN`.

Per visualizzare le sottoscrizioni attive a USA/Alabama e a tutti i relativi child, utilizzare il **runmqsc** comando `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN`.

Gestione del numero di argomenti in una struttura ad albero degli argomenti

In sintesi, esistono diversi modi per gestire il numero di argomenti in una struttura ad albero di argomenti.

Visualizza TPCOUNT

Utilizzare il **runmqsc** comando `DISPLAY PUBSUB ALL` periodicamente per visualizzare la proprietà **TPCOUNT**. Questo è il numero di nodi argomento nella struttura ad albero degli argomenti. Se il numero è in aumento, potrebbe indicare che è richiesto un TREELIFE più breve o che è necessaria una riprogettazione degli argomenti stessi.

Modificare TREELIFE

Per impostazione predefinita, un argomento inutilizzato ha una durata di 30 minuti. È possibile ridurre la durata di un argomento inutilizzato.

Ad esempio, il comando **runmqsc**, `ALTER QMGR TREELIFE(900)`, riduce la durata di un argomento non utilizzato da 30 minuti a 15 minuti.

Eccezionalmente, riavviare il gestore code

Quando il gestore code viene riavviato, la struttura ad albero degli argomenti viene reinizializzata da oggetti argomento, nodi con pubblicazioni conservate e sottoscrizioni durevoli. Gli argomenti che erano stati creati dal funzionamento dei programmi di pubblicazione e di sottoscrizione vengono eliminati.

Come ultima possibilità, se la crescita degli argomenti indesiderati è stata la causa di problemi di prestazioni nel passato, riavviare il gestore code.

Concetti correlati

[Strutture ad albero degli argomenti](#)

Aspera gateway può migliorare le prestazioni su reti ad alta latenza

IBM Aspera faspio Gateway fornisce un tunnel TCP/IP veloce che può aumentare in modo significativo la velocità di trasmissione di rete per IBM MQ.

Aspera gateway può essere utilizzato per migliorare le prestazioni dei canali del gestore code. È particolarmente efficace se la rete ha una latenza elevata o tende a perdere pacchetti ed è generalmente utilizzata per velocizzare la connessione tra gestori code in diversi data center.

Tuttavia, per una rete veloce che non perde pacchetti, le prestazioni diminuiscono quando si utilizza Aspera gateway, quindi è importante verificare le prestazioni della rete prima e dopo la definizione di una connessione Aspera gateway .

Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat® o Ubuntu Linux o Windows.

Per ulteriori informazioni, vedi [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti.

IBM potrebbe non offrire i prodotti, i servizi o le funzioni descritti in questo documento in altri paesi. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti da IBM possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM potrebbe disporre di applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura di tale documento non concede alcuna licenza a tali brevetti. Chi desiderasse ricevere informazioni relative a licenze può rivolgersi per iscritto a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali: INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe non essere applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche vengono incorporate nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti o modifiche al prodotto/i e/o al programma/i descritti nella pubblicazione in qualsiasi momento e senza preavviso.

Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti solo per consultazione e non rappresenta in alcun modo un'approvazione di tali siti. I materiali reperibili in tali siti Web non fanno parte dei materiali relativi a questo prodotto IBM e l'utilizzo di tali siti è responsabilità dell'utente.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente da IBM e diventeranno esclusiva della stessa.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Corporation
Coordinatore interoperabilità software, Dipartimento 49XA
Autostrada 3605 52 N

Rochester, MN 55901
U.S.A.

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in queste informazioni e tutto il materiale su licenza disponibile per esso sono forniti da IBM in base ai termini dell' IBM Customer Agreement, IBM International Program License Agreement o qualsiasi altro accordo equivalente tra le parti.

Tutti i dati relativi alle prestazioni contenuti in questo documento sono stati determinati in un ambiente controllato. Pertanto, i risultati ottenuti in altri ambienti operativi possono variare in modo significativo. Alcune misurazioni potrebbero essere state fatte su sistemi a livello di sviluppo e non vi è alcuna garanzia che queste misurazioni saranno le stesse sui sistemi generalmente disponibili. Inoltre, alcune misurazioni potrebbero essere state stimate mediante estrapolazione. I risultati quindi possono variare. Gli utenti di questo documento dovrebbero verificare i dati applicabili per il loro ambiente specifico.

Le informazioni relative a prodotti non IBM provengono dai fornitori di tali prodotti, dagli annunci pubblicati o da altre fonti pubblicamente disponibili. IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Tutte le dichiarazioni riguardanti la direzione o l'intento futuro di IBM sono soggette a modifica o ritiro senza preavviso e rappresentano solo scopi e obiettivi.

Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali. Per poterli illustrare nel modo più completo possibile, gli esempi riportano nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi adoperati da imprese realmente esistenti sono una mera coincidenza.

LICENZA SUL COPYRIGHT:

Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi.

Se si sta visualizzando queste informazioni in formato elettronico, le fotografie e le illustrazioni a colori potrebbero non apparire.

Informazioni sull'interfaccia di programmazione

Le informazioni sull'interfaccia di programmazione, se fornite, consentono di creare software applicativo da utilizzare con questo programma.

Questo manuale contiene informazioni sulle interfacce di programmazione che consentono al cliente di scrivere programmi per ottenere i servizi di IBM MQ.

Queste informazioni, tuttavia, possono contenere diagnosi, modifica e regolazione delle informazioni. La diagnosi, la modifica e la regolazione delle informazioni vengono fornite per consentire il debug del software applicativo.

Importante: Non utilizzare queste informazioni di diagnosi, modifica e ottimizzazione come interfaccia di programmazione poiché sono soggette a modifica.

Marchi

IBM, il logo IBM, ibm.com, sono marchi di IBM Corporation, registrati in molte giurisdizioni nel mondo. Un elenco aggiornato dei marchi IBM è disponibile sul web in "Copyright and trademark

information"www.ibm.com/legal/copytrade.shtml. Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o altre società.

Microsoft e Windows sono marchi di Microsoft Corporation negli Stati Uniti, in altri paesi o entrambi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e/o in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Questo prodotto include il software sviluppato da Eclipse Project (<https://www.eclipse.org/>).

Java e tutti i marchi e i logo Java sono marchi registrati di Oracle e/o di società affiliate.



Numero parte:

(1P) P/N: