

9.2

*Amministrazione di IBM MQ*

**IBM**

**Nota**

Prima di utilizzare queste informazioni e il prodotto supportato, leggere le informazioni in [“Informazioni particolari” a pagina 547](#).

Questa edizione si applica alla versione 9 release 2 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

<b>Amministrazione.....</b>	<b>7</b>
Modalità di gestione dei gestori code IBM MQ e delle risorse associate.....	8
Amministrazione di IBM MQ utilizzando i comandi di controllo.....	10
Gestione mediante comandi MQSC.....	11
Esecuzione interattiva dei comandi MQSC.....	13
Esecuzione dei comandi MQSC dai file di testo.....	17
Origini da cui è possibile emettere comandi MQSC su z/OS.....	20
Configurazione automatica da uno script MQSC all'avvio.....	21
Valori generici e caratteri con significati speciali.....	22
Automazione della gestione IBM MQ utilizzando i comandi PCF.....	23
Introduzione a IBM MQ Programmable Command Formats.....	24
Utilizzo di MQAI per semplificare l'utilizzo dei PCF.....	36
Amministrazione mediante REST API.....	73
Introduzione a administrative REST API.....	73
Gestione remota mediante REST API.....	78
REST API data/ora.....	82
REST API gestione degli errori.....	83
Rilevamento REST API.....	85
REST API supporto lingua nazionale.....	86
REST API versioni.....	88
Gestione mediante una console Web.....	90
Introduzione alla console Web.....	90
Presentazione rapida della nuova console Web.....	92
Passaggio da un tipo di console all'altro.....	112
Amministrazione mediante IBM MQ Explorer.....	114
Operazioni che è possibile eseguire con IBM MQ Explorer.....	114
Impostazione di IBM MQ Explorer.....	116
Utilizzo dell'applicazione IBM MQ Taskbar (solo Windows).....	122
L'applicazione di monitoraggio avvisi IBM MQ (solo Windows).....	122
Utilizzo degli oggetti IBM MQ locali.....	122
Uso dei gestori code.....	123
Arresto dei canali MQI.....	133
Gestione delle code locali.....	133
Gestione delle code remote.....	143
Gestione delle code alias.....	145
Utilizzo delle code modello.....	147
Gestione delle code di messaggi non recapitabili.....	148
Utilizzo degli argomenti di gestione.....	168
Utilizzo delle sottoscrizioni.....	171
Gestione dei servizi.....	175
Gestione degli oggetti per il trigger.....	182
Utilizzo del programma di utilità <b>dmpmqmsg</b> tra due sistemi.....	184
Utilizzo di oggetti IBM MQ remoti.....	187
Configurazione dei gestori code per la gestione remota.....	188
Gestione del server dei comandi per la gestione remota.....	192
Immissione di comandi MQSC su un gestore code remoto.....	193
Conversione dati tra serie di caratteri codificati.....	195
Amministrazione Managed File Transfer.....	199
Avvio di un agent MFT.....	200
Elenco di agenti di MFT.....	205
Arresto di un agent MFT.....	205
Avvio di un nuovo trasferimento file.....	206

Creazione di un trasferimento file pianificato.....	209
Gestione trasferimenti file in sospeso.....	210
Attivazione di un trasferimento file.....	211
Monitoraggio dei trasferimenti file in corso.....	212
Visualizzazione dello stato dei trasferimenti file nel log di trasferimento.....	215
Monitoraggio delle risorse MFT.....	217
Utilizzo dei modelli di trasferimento file.....	247
Trasferimento di dati dai file ai messaggi.....	250
Trasferimento dei dati dai messaggi ai file.....	258
Il bridge di protocollo.....	264
Il bridge Connect:Direct.....	285
Utilizzo di MFT da IBM Integration Bus.....	300
Ripristino e riavvio di MFT.....	300
Impostazione di una scadenza per il recupero di trasferimenti in stallo.....	301
AmministrazioneMQ Telemetry.....	306
Configurazione di un gestore code per la telemetria su Linux e AIX.....	307
Configurazione di un gestore code per la telemetria su Windows.....	309
Configurazione dell'accodamento distribuito per inviare messaggi ai client MQTT.....	311
Autenticazione, autorizzazione e identificazione del client MQTT.....	313
Autenticazione del canale di telemetria mediante TLS.....	319
Privacy di pubblicazione sui canali di telemetria.....	321
Configurazione TLS di canali di telemetria e client MQTT Java.....	322
Configurazione del canale di telemetria JAAS.....	326
Gestione di un client AMQP.....	328
Visualizzazione degli oggetti IBM MQ utilizzati dai clienti AMQP.....	328
Identificazione, autorizzazione e autenticazione del client AMQP.....	330
Privacy delle pubblicazioni sui canali.....	332
Configurazione dei client AMQP con TLS.....	332
Disconnessione dei client AMQP dal gestore code.....	333
Gestione multicast.....	334
Introduzione a multicast.....	334
Topologia argomento IBM MQ Multicast.....	335
Controllo della dimensione dei messaggi multicast.....	336
Abilitazione della conversione dati per la messaggistica multicast.....	338
Monitoraggio applicazione multicast.....	339
Affidabilità dei messaggi multicast.....	339
Attività multicast avanzate.....	340
AmministrazioneIBM MQ for IBM i.....	343
Gestione di IBM MQ for IBM i utilizzando i comandi CL.....	344
Modi alternativi di amministrazione di IBM MQ for IBM i.....	357
Gestione del lavoro per IBM i.....	363
Disponibilità, backup, ripristino e riavvio su IBM i.....	370
In corso di quiesceIBM MQ for IBM i.....	414
AmministrazioneIBM MQ for z/OS.....	417
Immissione di comandi in IBM MQ for z/OS.....	418
I programmi di utilità IBM MQ for z/OS.....	426
Funzionamento di IBM MQ for z/OS.....	429
Scrittura di programmi per amministrare IBM MQ for z/OS.....	450
Gestione delle risorse IBM MQ su z/OS.....	462
Ripristino e riavvio su z/OS.....	501
IBM MQ e IMS.....	523
Funzionamento di Advanced Message Security su z/OS.....	536
AmministrazioneIBM MQ Internet Pass-Thru.....	537
avvio e arrestoMQIPT.....	537
Gestione di MQIPT utilizzando la riga comandi.....	539
Esecuzione di backup.....	545
Ottimizzazione delle prestazioni.....	546

<b>Informazioni particolari.....</b>	<b>547</b>
Informazioni sull'interfaccia di programmazione.....	548
Marchi.....	548



# Amministrazione IBM MQ

---

Per gestire i gestori code IBM MQ e le risorse associate, scegliere il metodo preferito da una serie di attività che è possibile utilizzare per attivare e gestire tali risorse.

## Informazioni su questa attività

È possibile gestire gli oggetti IBM MQ in locale o in remoto:

### gestione locale

Per amministrazione locale si intende l'esecuzione di attività di amministrazione su qualsiasi gestore code definito sul sistema locale. È possibile accedere ad altri sistemi, ad esempio tramite il programma di emulazione del terminale TCP/IP **telneted** eseguire la gestione. In IBM MQ, è possibile considerarlo come amministrazione locale poiché non sono coinvolti canali, ossia la comunicazione è gestita dal sistema operativo.

Per ulteriori informazioni, consultare [“Utilizzo degli oggetti IBM MQ locali”](#) a pagina 122.

### Amministrazione remota

IBM MQ supporta l'amministrazione da un singolo punto di contatto tramite amministrazione remota. La gestione remota consente di immettere comandi dal sistema locale che vengono elaborati su un altro sistema e si applica anche a IBM MQ Explorer. Ad esempio, è possibile immettere un comando remoto per modificare una definizione di coda su un gestore code remoto. Non è necessario accedere a tale sistema, anche se è necessario definire i canali appropriati. Il gestore code e server dei comandi sul sistema di destinazione devono essere in esecuzione.

Alcuni comandi non possono essere emessi in questo modo, in particolare, creando o avviando i gestori code e avviando i server dei comandi. Per eseguire questo tipo di attività, è necessario accedere al sistema remoto e immettere i comandi da lì o creare un processo che possa emettere i comandi per l'utente. Questa limitazione si applica anche a IBM MQ Explorer.

Per ulteriori informazioni, consultare [“Utilizzo di oggetti IBM MQ remoti”](#) a pagina 187.

Esistono diversi metodi che è possibile utilizzare per creare e gestire i gestori code e le risorse correlate in IBM MQ. Questi metodi includono interfacce della riga comandi, interfacce utente grafiche e un'API di gestione.

Esistono diverse serie di comandi che è possibile utilizzare per gestire IBM MQ a seconda della piattaforma:

- [“Comandi di controllo IBM MQ”](#) a pagina 8
- [“Comandi di script IBM MQ \(MQSC\)”](#) a pagina 8
- [“PCF \(Programmable Command Format\)”](#) a pagina 9
- [administrative REST API](#)
-  [“CL \(Control Language\) IBM i”](#) a pagina 9

Ci sono anche le seguenti opzioni per la creazione e la gestione di oggetti IBM MQ :

-   [“Il IBM MQ Explorer”](#) a pagina 9
- [“Il IBM MQ Console”](#) a pagina 10
-  [“MCS \( Microsoft Cluster Service\)”](#) a pagina 10

 [z/OS](#) Per informazioni sulle interfacce e le opzioni di gestione su IBM MQ for z/OS, consultare [“Amministrazione IBM MQ for z/OS”](#) a pagina 417.

È possibile automatizzare alcune attività di gestione e monitoraggio sia per i gestori code locali che remoti utilizzando i comandi PCF. Questi comandi possono essere semplificati anche utilizzando MQAI ( IBM MQ

Administration Interface) su alcune piattaforme. Per ulteriori informazioni sull'automazione delle attività di gestione, consultare [“Automazione della gestione IBM MQ utilizzando i comandi PCF”](#) a pagina 23.

### Concetti correlati

[IBM MQ - Sommario tecnico](#)

### Attività correlate

[Pianificazione](#)

[Configurazione](#)

### Riferimenti correlati

[Confronto serie di comandi](#)

## Modalità di gestione dei gestori code IBM MQ e delle risorse associate

---

Esistono diverse opzioni per la gestione dei gestori code IBM MQ e delle risorse associate.

### Comandi di controllo IBM MQ

**ALW**

È possibile eseguire attività di gestione sui gestori code stessi utilizzando i comandi di controllo.

I sistemi IBM MQ for AIX, Linux®, and Windows forniscono i *comandi di controllo* immessi dalla riga comandi del sistema.

I comandi di controllo sono descritti in [Creazione e gestione di gestori code su Multiplatforms](#). Per il riferimento ai comandi per i comandi di controllo, consultare [IBM MQ Comandi di controllo](#).

### Comandi di script IBM MQ (MQSC)

Utilizzare i comandi MQSC per gestire gli oggetti del gestore code, inclusi il gestore code, le code, le definizioni di processo, gli elenchi nomi, i canali, i canali di connessione client, i listener, i servizi e gli oggetti delle informazioni di autenticazione.

Si immettono comandi MQSC a un gestore code utilizzando il comando **runmqsc**. È possibile eseguire questa operazione in modo interattivo, immettendo i comandi da una tastiera oppure reindirizzare l'unità di immissione standard (stdin) per eseguire una sequenza di comandi da un file di testo ASCII. In entrambi i casi, il formato dei comandi è lo stesso.

È possibile eseguire il comando **runmqsc** in tre modalità, a seconda degli indicatori impostati sul comando:

- *Modalità di verifica*, in cui i comandi MQSC vengono verificati su un gestore code locale, ma non vengono eseguiti.
- *Modalità diretta*, in cui i comandi MQSC vengono eseguiti su un gestore code locale.
- *Modalità indiretta*, dove i comandi MQSC vengono eseguiti su un gestore code remoto.

I comandi MQSC sono disponibili su tutte le piattaforme, inclusi IBM i e z/OS. I comandi MQSC sono riepilogati in [Confronto delle serie comandi](#).

**ALW**

Su AIX, Linux, and Windows, è possibile utilizzare MQSC come singoli comandi emessi sulla riga comandi del sistema. Per emettere comandi più complessi o multipli, è possibile creare MQSC in un file che si esegue dalla riga comandi. I comandi MQSC possono essere inviati a un gestore code remoto. Per i dettagli completi, consultare [“Esecuzione dei comandi MQSC dai file di testo”](#) a pagina 17.

**IBM i**

Per immettere i comandi su un server IBM i, creare un elenco di comandi in un file di script, quindi eseguire il file utilizzando il comando STRMQMMQSC.

**Note:** **IBM i**

1. Non utilizzare la libreria QTEMP come libreria di input per STRMQMMQSC, poiché l'utilizzo della libreria QTEMP è limitato. È necessario utilizzare un'altra libreria come file di immissione per il comando.
  2. Su IBM i, le risposte MQSC ai comandi emessi da un file script vengono restituite in un file di spool.
- Per ulteriori informazioni sull'utilizzo dei comandi MQSC, consultare [“Gestione mediante comandi MQSC”](#) a pagina 11.

## PCF (Programmable Command Format)

I PCF (Programmable Command Format) definiscono i messaggi di comando e risposta che possono essere scambiati tra un programma e qualsiasi gestore code (che supporta i PCF) in una rete. È possibile utilizzare i comandi PCF in un programma di applicazione di gestione dei sistemi per la gestione degli oggetti IBM MQ : oggetti delle informazioni di autenticazione, canali, listener dei canali, elenchi nomi, definizioni di processi, gestori code, code, servizi e classi di memoria. L'applicazione può operare da un singolo punto nella rete per comunicare le informazioni di comando e risposta con qualsiasi gestore code, locale o remoto, utilizzando il gestore code locale.

Per ulteriori informazioni sui PCF, consultare [“Introduzione a IBM MQ Programmable Command Formats”](#) a pagina 24.

Per la definizione di PCF e strutture per comandi e risposte, consultare [Programmable command formats reference](#).

## Il administrative REST API

administrative REST API fornisce un'interfaccia RESTful che è possibile utilizzare per gestire IBM MQ. Quando si utilizza administrative REST API, si richiama un metodo HTTP su un URL che rappresenta un oggetto IBM MQ . Ad esempio, è possibile richiedere informazioni sulle installazioni di IBM MQ utilizzando il metodo HTTP GET sul seguente URL:

```
https://localhost:9443/ibmmq/rest/v1/admin/installation
```

È possibile utilizzare administrative REST API con l'implementazione HTTP/REST di un linguaggio di programmazione o utilizzando strumenti come cURL o un componente aggiuntivo del browser del client REST.

Per ulteriori informazioni, consultare [administrative REST API](#)

## CL (Control Language) IBM i



Questa lingua può essere utilizzata per immettere comandi di amministrazione in IBM MQ for IBM i. I comandi possono essere immessi sulla riga comandi o scrivendo un programma CL. Questi comandi eseguono funzioni simili ai comandi PCF, ma il formato è diverso. I comandi CL sono progettati esclusivamente per i server e le risposte CL sono leggibili, mentre i comandi PCF sono indipendenti dalla piattaforma e sia i formati comando che risposta sono destinati all'utilizzo del programma.

Per informazioni dettagliate su IBM i CL (Control Language), consultare [IBM MQ for IBM i Comandi CL](#).

## Il IBM MQ Explorer



Utilizzando IBM MQ Explorer, è possibile eseguire le seguenti azioni:

- Definire e controllare varie risorse, come gestori code, code, definizioni di processi, elenchi nomi, canali, canali di connessione client, listener, servizi e cluster.
- Avviare o arrestare un gestore code locale e i relativi processi associati.
- Visualizzare i gestori code e i relativi oggetti associati sulla stazione di lavoro o da altre stazioni di lavoro.
- Verificare lo stato dei gestori code, dei cluster e dei canali.

- Verificare quali applicazioni, utenti o canali hanno una particolare coda aperta, dallo stato della coda.

Su sistemi Windows e Linux , è possibile avviare IBM MQ Explorer utilizzando il menu di sistema, il file eseguibile MQExpLorer o il comando **strmqcfcg** .

#### Linux

Su Linux, per avviare correttamente IBM MQ Explorer , è necessario essere in grado di scrivere un file nella propria directory home e la directory home deve esistere.

Per ulteriori informazioni, consultare [“Amministrazione mediante IBM MQ Explorer”](#) a pagina 114.

È possibile utilizzare IBM MQ Explorer per gestire i gestori code remoti su altre piattaforme, incluso z/OS.

IBM MQ Explorer può essere installato come parte dell'installazione del prodotto (consultare [Installazione e disinstallazione di IBM MQ](#)) oppure dal download IBM MQ Explorer autonomo disponibile da Fix Central (consultare [Installazione e disinstallazione di IBM MQ Explorer come applicazione autonoma su Linux e Windows](#)).

## Il IBM MQ Console

È possibile utilizzare IBM MQ Console per amministrare IBM MQ da un browser Web.

Per ulteriori informazioni, consultare [“Gestione mediante una console Web”](#) a pagina 90.

## MSCS ( Microsoft Cluster Service)

#### Windows

Microsoft Cluster Service (MSCS) consente di collegare i server in un *cluster*, fornendo una maggiore disponibilità di dati e applicazioni e rendendo più facile la gestione del sistema. MSCS è in grado di rilevare e ripristinare automaticamente gli errori del server o dell'applicazione.

È importante non confondere i cluster nel senso MSCS con i cluster IBM MQ . La distinzione è la seguente:

#### IBM MQCluster

Si tratta di gruppi di due o più gestori code su uno o più computer, che forniscono l'interconnessione automatica e consentono la condivisione delle code per il bilanciamento del carico e la ridondanza.

#### Cluster MSCS

Si tratta di gruppi di computer, che sono collegati tra loro e configurati in modo che, in caso di errore, MSCS esegua un *failover*, trasferendo i dati di stato delle applicazioni dal computer in errore a un altro computer nel cluster e inizializzandone nuovamente l'operazione.

[Supporto del servizio cluster Microsoft \(MSCS\)](#) fornisce informazioni dettagliate su come configurare il tuo sistema IBM MQ for Windows per utilizzare MSCS.

#### ALW

## Amministrazione di IBM MQ utilizzando i comandi di controllo

I comandi di controllo forniscono un modo per eseguire una serie di attività di gestione IBM MQ su AIX, Linux, and Windows.

Se si desidera immettere comandi di controllo, l'ID utente deve essere un membro del gruppo mqm per la maggior parte dei comandi di controllo. Per ulteriori informazioni su questo argomento, consultare [Authority to amministrare IBM MQ su AIX, Linux, and Windows](#). Inoltre, prendere nota delle informazioni specifiche dell'ambiente. per la piattaforma o le piattaforme utilizzate dall'azienda.

Quando si utilizzano i comandi di controllo che operano su un gestore code, è necessario utilizzare il comando dell'installazione associata al gestore code che si sta gestendo.

Quando si utilizzano comandi di controllo che operano su un gestore code configurato per utilizzare l'autenticazione della connessione con CHCKLOCL (REQUIRED) e si osserva un errore di connessione,

- Fornire un ID utente e una parola d'ordine se il comando di controllo lo consente.
- Utilizzare gli equivalenti MQSC dei comandi di controllo laddove esistono.

- Avviare il gestore code utilizzando l'opzione `-ns`, mentre è necessario eseguire i comandi di controllo che non possono connettersi.

Per un elenco completo dei comandi di controllo, consultare [Comandi di controllo IBM MQ](#).

## Utilizzo dei comandi di controllo su sistemi Windows

### Windows

In IBM MQ for Windows, immettere i comandi di controllo da un prompt dei comandi.

I comandi di controllo e i relativi indicatori non sono sensibili al maiuscolo / minuscolo, ma gli argomenti per tali comandi, come i nomi delle code e dei gestori code, sono sensibili al maiuscolo / minuscolo.

Ad esempio, nel comando:

```
crtmqm /u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- Il nome del comando può essere immesso in lettere maiuscole o minuscole oppure in una combinazione delle due. Questi sono tutti validi: `crtmqm`, `CRTMQM`, `CRTmqm`.
- L'indicatore può essere immesso come `-u`, `-U`, `/uo` /`U`.
- `SYSTEM.DEAD.LETTER.QUEUE` e `jupiter.queue.manager` devono essere immessi esattamente come mostrato.

## Utilizzo dei comandi di controllo su sistemi AIX and Linux

### Linux AIX

Nei sistemi IBM MQ for AIX or Linux , immettere i comandi di controllo in una finestra della shell.

Negli ambienti UNIX and Linux , i comandi di controllo, incluso il nome del comando stesso, gli indicatori e gli eventuali argomenti, sono sensibili al maiuscolo / minuscolo. Ad esempio, nel comando:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- Il nome del comando deve essere `crtmqm`, non `CRTMQM`.
- L'indicatore deve essere `-u`, non `-U`.
- La coda di messaggi non recapitabili è denominata `SYSTEM.DEAD.LETTER.QUEUE`.
- L'argomento è specificato come `jupiter.queue.manager`, che è diverso da `JUPITER.queue.manager`.

Fare attenzione a digitare i comandi esattamente come vengono visualizzati negli esempi.

### Riferimenti correlati

[IBM MQ Riferimento comandi di controllo](#)

## Gestione mediante comandi MQSC

È possibile utilizzare i comandi MQSC per gestire gli oggetti gestore code, inclusi il gestore code stesso, le code, le definizioni di processo, canali, canali di connessione client, listener, servizi, elenchi nomi, cluster e oggetti delle informazioni di autenticazione. I comandi MQSC sono disponibili su tutte le piattaforme.

### Informazioni su questa attività

Il modo in cui si eseguono i comandi MQSC dipende dalla piattaforma:

- **ALW** Su AIX, Linux, and Windows, si immettono comandi MQSC a un gestore code utilizzando il comando `runmqsc` . È possibile eseguire il comando `runmqsc` in diversi modi:

- In modo interattivo, immettendo comandi da una tastiera. Consultare [“Esecuzione interattiva dei comandi MQSC”](#) a pagina 13.
- Da un file di testo ASCII. Consultare [“Esecuzione dei comandi MQSC dai file di testo”](#) a pagina 17.
- Su un gestore code remoto. Consultare [“Utilizzo di oggetti IBM MQ remoti”](#) a pagina 187.
- **z/OS** Su z/OS, i comandi MQSC possono essere emessi da un numero di origini, a seconda del comando. Per ulteriori informazioni, consultare [“Origini da cui è possibile emettere comandi MQSC su z/OS”](#) a pagina 20.

I comandi MQSC sono dettagliati nella sezione [Comandi MQSC](#) .

## Procedura

- Ogni comando inizia con un parametro primario (un verbo) e questo è seguito da un parametro secondario (un nome). Questo è quindi seguito dal nome o dal nome generico dell'oggetto (tra parentesi) se ne esiste uno, che è presente nella maggior parte dei comandi. Successivamente, i parametri possono di solito verificarsi in qualsiasi ordine; se un parametro ha un valore corrispondente, il valore deve essere immediatamente successivo al parametro a cui si riferisce.

**Nota:** **z/OS** Su z/OS, il parametro secondario non deve essere il secondo.

- Le parole chiave, le parentesi e i valori possono essere separati da qualsiasi numero di spazi e virgole. Una virgola mostrata nei diagrammi di sintassi può essere sempre sostituita da uno o più spazi vuoti. Ogni parametro deve essere preceduto da almeno uno spazio vuoto (dopo il parametro primario) tranne che su z/OS .
- Qualsiasi numero di spazi vuoti può verificarsi all'inizio o alla fine del comando e tra parametri, punteggiatura e valori. Ad esempio, è valido il seguente comando:

```
ALTER QLOCAL ('Account' ) TRIGDPTH ( 1)
```

Gli spazi vuoti all'interno di una coppia di apici sono significativi.

- Le virgole aggiuntive possono essere visualizzate in qualsiasi punto in cui sono consentiti spazi e vengono trattate come se fossero spazi vuoti (a meno che, naturalmente, non siano all'interno di stringhe racchiuse tra virgolette).
- Non sono consentiti parametri ripetuti. La ripetizione di un parametro con la sua versione "NO", come in REPLACE NOREPLACE, non è consentita.
- Le stringhe che contengono spazi, caratteri minuscoli o caratteri speciali devono essere racchiuse tra virgolette singole, a meno che non si verifichi una delle seguenti condizioni:
  - I caratteri speciali sono uno o più dei seguenti:
    - Punto (.)
    - Barra (/)
    - Trattino basso (\_)
    - Segno percentuale (%)
  - **z/OS** Il comando viene emesso dalle operazioni IBM MQ for z/OS e dai pannelli di controllo.
  - La stringa è un valore generico che termina con un asterisco. (su IBM i devono essere racchiuse tra virgolette singole)
  - La stringa è un singolo asterisco, ad esempio TRACE (\*) (su IBM i devono essere racchiusi tra virgolette singole)
  - La stringa è una specifica di intervallo contenente due punti, ad esempio CLASS (01:03)

Se la stringa contiene una virgoletta singola, la virgoletta singola è rappresentata da due virgolette singole. I caratteri minuscoli non contenuti tra virgolette vengono ripiegati in maiuscolo.

- **Multi**

Su **Multiplatforms**, una stringa che non contiene caratteri (ossia, due virgolette singole senza spazi tra di loro) viene interpretata come uno spazio vuoto racchiuso tra virgolette singole, vale a dire, interpretato nello stesso modo di ("). L'eccezione è se l'attributo utilizzato è uno dei seguenti attributi, quando due virgolette singole senza spazio vengono interpretate come una stringa di lunghezza zero:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

-  **z/OS**

Su **z/OS**, se si desidera uno spazio vuoto racchiuso tra virgolette singole, è necessario immetterlo come tale ("). Una stringa che non contiene caratteri (") è uguale all'immissione di ().

- Gli spazi vuoti finali negli attributi stringa basati sui tipi MQCHARV, come ad esempio SELECTOR, dati utente secondari, vengono trattati come significativi, il che significa che ' abc ' non è uguale a ' abc '.
- Una parentesi di apertura seguita da una parentesi di chiusura, senza alcuna informazione significativa nel mezzo, non è valida se non dove specificatamente indicato. Ad esempio, la seguente stringa non è valida:

```
NAME ( )
```

- Le parole chiave non sono sensibili al maiuscolo / minuscolo: AltER, alter e ALTER sono tutte accettabili. Tutto ciò che non è contenuto tra virgolette viene ripiegato in maiuscolo.
- I sinonimi sono definiti per alcuni parametri. Ad esempio, DEF è sempre un sinonimo di DEFINE, quindi DEF QLOCAL è valido. I sinonimi non sono, tuttavia, solo stringhe minime; DEFI non è un sinonimo valido per DEFINE.

**Nota:** Non esiste alcun sinonimo per il parametro DELETE. Ciò per evitare l'eliminazione accidentale di oggetti quando si utilizza DEF, il sinonimo di DEFINE.

- I comandi MQSC utilizzano determinati caratteri speciali per avere determinati significati. Per ulteriori informazioni su questi caratteri speciali e su come utilizzarli, consultare [“Valori generici e caratteri con significati speciali”](#) a pagina 22.

### Attività correlate

[Risoluzione dei problemi con i comandi MQSC](#)

### Riferimenti correlati

[runmqsc \(esecuzione comandi MQSC\)](#)

## Esecuzione interattiva dei comandi MQSC

È possibile utilizzare i comandi MQSC in modo interattivo utilizzando una finestra di comandi o una shell.

### Prima di iniziare

È possibile impostare il prompt visualizzato quando si esegue il comando **runmqsc**. Consultare [“Impostazione del prompt dei comandi MQSC”](#) a pagina 15.

  Quando si eseguono i comandi MQSC in modo interattivo su piattaforme UNIX and Linux, la linea di comandi **runmqsc** supporta il richiamo dei comandi, il completamento dei comandi e le chiavi di comando Emacs. Consultare [“Il comando runmqsc su AIX e Linux.”](#) a pagina 16.

### Informazioni su questa attività

## Procedura

1. Per utilizzare i comandi MQSC in modo interattivo, aprire una finestra di comandi o una shell e immettere il seguente comando:

```
runmqsc QMgrName
```

Dove *QMgrName* specifica il nome del gestore code che si desidera elaborare i comandi MQSC. È possibile lasciare vuoto il campo *QMgrName* per elaborare comandi MQSC sul gestore code predefinito.

2. Immettere i comandi MQSC, come richiesto. Ad esempio, per creare una coda locale denominata ORANGE . LOCAL . QUEUE immettere il seguente comando:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Per i comandi che hanno troppi parametri per adattarsi su una riga, utilizzare i caratteri di continuazione per indicare che un comando continua sulla seguente riga:

- Un segno meno (-) indica che il comando deve continuare dall'inizio della seguente riga.
- Un segno più (+) indica che il comando deve continuare dal primo carattere non vuoto sulla seguente riga.

L'immissione del comando termina con il carattere finale di una riga non vuota che non è un carattere di continuazione. È anche possibile terminare esplicitamente l'input del comando immettendo un punto e virgola (;).

3. Interrompere l'utilizzo dei comandi MQSC immettendo il comando seguente:

```
end
```

In alternativa, è possibile utilizzare il carattere EOF per il proprio sistema operativo.

## Risultati

Quando si immettono i comandi MQSC, il gestore code restituisce i messaggi dell'operatore che confermano le azioni dell'utente o che indicano gli errori effettuati. Ad esempio, il seguente messaggio conferma la creazione di una coda:

```
AMQ8006: IBM MQ queue created.
```

Il seguente messaggio indica che è stato commesso un errore di sintassi:

```
AMQ8405: Syntax error detected at or near end of command segment below:-  
AMQ8426: Valid MQSC commands are:
```

```
ALTER  
CLEAR  
DEFINE  
DELETE  
DISPLAY  
END  
PING  
REFRESH  
RESET  
RESOLVE  
RESUME  
START  
STOP  
SUSPEND  
4 : end
```

Questi messaggi vengono inviati alla periferica di output standard. Se il comando non è stato immesso correttamente, fare riferimento alle informazioni di riferimento del comando per trovare la sintassi corretta. Vedere [Comandi MQSC](#).

### Attività correlate

[“Esecuzione dei comandi MQSC dai file di testo” a pagina 17](#)

L'esecuzione interattiva dei comandi MQSC è adatta per test rapidi, ma se si dispone di comandi molto lunghi o si utilizza una particolare sequenza di comandi ripetutamente, è possibile reindirizzare stdin da un file di testo. È anche possibile reindirizzare l'output a un file.

### Riferimenti correlati

[runmqsc](#)

## Impostazione del prompt dei comandi MQSC

È possibile impostare il prompt dei comandi MQSC su un prompt a scelta utilizzando la variabile di ambiente **MQPROMPT**.

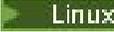
### Procedura

- Impostare la variabile di ambiente **MQPROMPT** sulla richiesta desiderata.

Il prompt viene inserito sia quando la console **runmqsc** viene eseguita in modo interattivo, sia quando l'input viene reindirizzato in **runmqsc** da un file o dal dispositivo di input standard (stdin).

È possibile includere testo semplice nel prompt dei comandi ed è anche possibile inserire variabili di ambiente utilizzando la notazione +VARNAME+ nello stesso modo delle definizioni di oggetto servizio IBM MQ. Per ulteriori informazioni, consultare [“Inserimenti sostituibili sulle definizioni di servizio” a pagina 178](#).

Esistono diversi altri inserimenti sostituibili aggiuntivi forniti da IBM MQ, descritti nella seguente tabella.

Inserimento sostituibile	Descrizione
NOME MQ_HOST_	Nome host del sistema
SEP FILE MQ	Separatore file specifico della piattaforma: –   Su sistemi AIX and Linux , MQ_FILE_SEP è /. –  Su sistemi Windows , l'ubicazione di MQ_FILE_SEP è \
MQ_PATH_SEP	Separatore di percorso specifico della piattaforma: –   Su sistemi AIX and Linux , MQ_PATH_SEP è :. –  Su sistemi Windows , l'ubicazione di MQ_PATH è ;
ORA_DATA_MQ	Data e ora del sistema locale in un formato YYYY-MM-DD hh:mm:ss.SSS fisso, ad esempio: <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;">2020-12-25 17:41:37.408</div>

### Note:

- I valori degli inserimenti sostituibili MQ si riferiscono all'installazione IBM MQ e al sistema host a cui è associato il comando **runmqsc**.

- MQPROMPT è limitato ad un massimo di 256 caratteri quando gli inserimenti vengono espansi. Le espansioni MQPROMPT su questo valore determinano il troncamento dell'intera stringa MQPROMPT senza le espansioni.

Ad esempio, per impostare il prompt su MQSC, immettere uno dei seguenti comandi:

- **Windows**

```
set "MQPROMPT=MQSC"
```

- **Linux** **AIX**

```
export MQPROMPT="MQSC"
```

## Esempio

Il seguente esempio mostra l'impostazione della variabile **MQPROMPT** su un sistema AIX . Il prompt è impostato per visualizzare un nome utente, preso dalla variabile di ambiente di sistema associata, il nome del gestore code e il nome host IBM MQ preso dagli inserimenti sostituibili di MQ :

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
sh> runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
C:\ > runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Il seguente esempio aggiunge una data / ora agli esempi MQPROMPT precedenti, ricavati dagli inserimenti sostituibili MQ :

```
sh> export MQPROMPT="+MQ_DATE_TIME+ +USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
sh> runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

2020-11-24 18:10:00.404 myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=MQ_DATE_TIME+ +USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "
C:\ > runmqsc MY.QMGR
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager MY.QMGR.

2020-11-24 18:10:01.007 myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

- **Linux**

- **AIX**

## Il comando **runmqsc** su AIX e Linux .

La riga di comando **runmqsc** su AIX e Linux supporta il comando di richiamo, il completamento del comando e i tasti di comando di Emacs.

Sono disponibili le seguenti funzioni dell'editor della riga comandi:

- Richiamo dei comandi immessi in precedenza utilizzando il tasto freccia su e il tasto freccia giù
- Completamento automatico per la parola chiave successiva di un comando utilizzando il tasto di tabulazione e la barra spaziatrice
- Tasti di comando Emacs o funzioni di tasti di comando simili

Per utilizzare queste funzioni, è necessario installare la libreria curses. Se la libreria curses non è installata sul proprio sistema, **runmqsc** non dispone delle funzioni dell'editor della riga comandi e viene

visualizzato un messaggio quando viene avviata la riga di comando **runmqsc** . Il nome della libreria curses da installare dipende dalla piattaforma UNIX :

-  Su AIX, installare curses.
- Su Linux, installare ncurses.

## Installazione di ncurses o curses su AIX

**Nota:** Il seguente esempio utilizza le istruzioni per Linux

Immettere il seguente comando per individuare i package ncurses esistenti:

```
rpm -qa | grep -i ncurses
```

I pacchetti ncurses richiesti sono i seguenti:

```
ncurses-term-6.1-7.20180224.el8.noarch
ncurses-6.1-7.20180224.el8.x86_64
ncurses-base-6.1-7.20180224.el8.noarch
ncurses-c++-libs-6.1-7.20180224.el8.x86_64
ncurses-libs-6.1-7.20180224.el8.x86_64
ncurses-compat-libs-6.1-7.20180224.el8.x86_64
ncurses-devel-6.1-7.20180224.el8.x86_64
```

È possibile installare tutti i package ncurses richiesti elencati nel testo precedente eseguendo il seguente comando:

```
yum install ncurses*
```

## Personalizzazione delle associazioni tasti Emacs

È possibile personalizzare le chiavi associate ai comandi. Ad esempio, è possibile collegare le chiavi alle associazioni vi invece delle associazioni di tasti Emacs predefinite.

Le chiavi vengono personalizzate modificando il file `.editrc` memorizzato nella directory home. Per ulteriori informazioni, consultare [editrc](#) nelle pagine man FreeBSD .

## Disabilitazione del richiamo del comando, del completamento del comando e dei tasti di comando Emacs

È possibile disabilitare il richiamo del comando, il completamento del comando e i tasti comando Emacs impostando una variabile di ambiente. Impostare la variabile di ambiente **MQ\_OVERRIDE\_LIBEDIT\_LOAD** su TRUE.

Questa variabile di ambiente può essere utilizzata come soluzione temporanea quando **runmqsc** mostra il seguente messaggio informativo:

```
AMQ8521I: Command completion and history unavailable
```

## Esecuzione dei comandi MQSC dai file di testo

L'esecuzione interattiva dei comandi MQSC è adatta per test rapidi, ma se si dispone di comandi molto lunghi o si utilizza una particolare sequenza di comandi ripetutamente, è possibile reindirizzare `stdin` da un file di testo. È anche possibile reindirizzare l'output a un file.

## Informazioni su questa attività

L'input per il comando **runmqsc** viene preso dalla *periferica di input standard*, indicata anche come `stdin`. `stdin` è l'unità da cui viene preso l'input per il sistema. In genere questa è la tastiera, ma è possibile specificare che l'input deve provenire da una porta seriale o da un file disco, ad esempio.

L'output per il comando **runmqsc** viene emesso nella *unità di output standard*, indicata anche come stdout. stdout è l'unità a cui viene inviato l'output dal sistema. In genere si tratta di una visualizzazione, ma è possibile reindirizzare l'output a una porta seriale o a un file.

È possibile creare i comandi MQSC in uno script quando si utilizzano:

- ▶ **z/OS** I dataset di inizializzazione CSQINP1, CSQINP2e CSQINPX o il programma di utilità batch CSQUTIL su z/OS.
- ▶ **IBM i** Il comando **STRMQM** su IBM i.
- ▶ **ALW** Il comando **runmqsc** su AIX, Linux, and Windows.

È possibile impostare il prompt dei comandi MQSC su un prompt a scelta utilizzando la variabile di ambiente **MQPROMPT**. Per ulteriori informazioni, consultare [“Impostazione del prompt dei comandi MQSC” a pagina 15](#).

## Procedura

1. Creare un file di testo che contenga i comandi MQSC che si desidera eseguire.

- Per la portabilità tra ambienti IBM MQ, limitare la lunghezza della riga nei file di comandi MQSC a 72 caratteri.
- Ciascun comando deve iniziare su una nuova riga.
- Una riga che inizia con un asterisco (\*) nella prima posizione viene ignorata. Questo può essere utilizzato per inserire commenti nel file.
- Le righe vuote vengono ignorate.
- Un segno più (+) indica che il comando continua dal primo carattere non vuoto nella riga successiva. Se si utilizza + per continuare un comando, ricordarsi di lasciare almeno uno spazio vuoto prima del parametro successivo (tranne su z/OS dove non è necessario). I commenti o le righe vuote vengono eliminati quando il comando viene riassemblato in una singola stringa.
- Un segno meno (-), indica che il comando deve essere continuato dall'inizio della riga successiva. I commenti o le righe vuote vengono eliminati quando il comando viene riassemblato in una singola stringa.
- I comandi MQSC contenuti in un comando Escape PCF (Programmable Command Format) non possono continuare con il segno più o con il segno meno. L'intero comando deve essere contenuto in un singolo comando Escape. Per informazioni sui comandi PCF, consultare [“Introduzione a IBM MQ Programmable Command Formats” a pagina 24](#).
- Su [Multiplatformse](#) su z/OS per i comandi emessi dal programma di utilità batch CSQUTIL, è possibile utilizzare un carattere punto e virgola (;) per terminare un comando, anche se è stato immesso un segno più (+) alla fine della riga precedente.
- Una riga non deve terminare con un carattere di controllo della tastiera (ad esempio, una tabulazione).
- Se si esegue il comando **runmqsc** in modalità client reindirizzando stdin da un file di testo e si fornisce l'indicatore **-u** per fornire le credenziali, il comando **runmqsc** non richiede una password e la password viene letta da stdin. Verificare che la prima riga di dati fornita tramite stdin sia la password. Questa operazione può essere eseguita utilizzando strumenti della riga comandi come "echo" o "cat" e passando la password seguita dallo script MQSC nel **runmqsc** comando stdin.
- ▶ **Windows** Su Windows, se alcuni caratteri speciali come il segno cancelletto (£) e il NOT logico (¬) vengono utilizzati in uno script di comandi (ad esempio, come parte di una descrizione dell'oggetto), vengono visualizzati in modo diverso nell'output di un comando come **DISPLAY QLOCAL**.
- Per informazioni sulla sintassi del comando MQSC, consultare [Comandi MQSC](#).
- È possibile utilizzare i file di comando MQSC di esempio per creare il proprio file di testo:

### **amqscos0.tst**

Definizioni di oggetti utilizzati da programmi di esempio.

## amqscic0.tst

Definizioni di code per transazioni CICS .

**Windows** Su Windows, questi file si trovano nella directory `MQ_INSTALLATION_PATH\tools\mqsc\samples`. `MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ .

**Linux** **AIX** Su AIX and Linux, questi file si trovano nella directory `MQ_INSTALLATION_PATH/samp`. `MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ .

2. Verificare che la sintassi del comando sia corretta su un gestore code locale senza eseguire i comandi. Utilizzare l'indicatore **-v** sul comando **runmqsc** .

**V 9.2.0**

- Da IBM MQ 9.2.0, utilizzare l'opzione **-f** per identificare il nome file del testo di input. Ad esempio:

```
runmqsc -f myprog.in -v QmgrName
```

- Per le release Long Term Support precedenti a IBM MQ 9.2.0e le release Continuous Delivery precedenti a IBM MQ 9.1.4, utilizzare l'operatore **<** per indirizzare i comandi MQSC dal file di testo di input al comando. Ad esempio:

```
runmqsc -v QmgrName < myprog.in
```

Il report restituito è simile a quello mostrato in [Figura 2 a pagina 20](#).

Non è possibile specificare un gestore code remoto durante la verifica dei comandi. Ciò significa che non è possibile specificare l'indicatore **-w** .

3. Quando la sintassi del comando è corretta, rimuovere l'indicatore **-v** ed eseguire nuovamente il comando **runmqsc** .

**V 9.2.0**

- Da IBM MQ 9.2.0 eseguire (ad esempio) il seguente comando:

```
runmqsc -f myprog.in QmgrName
```

- Per le release Long Term Support precedenti a IBM MQ 9.2.0e le release Continuous Delivery precedenti a IBM MQ 9.1.4, utilizzare uno dei seguenti comandi:
  - L'operatore **<** dirige l'input da un file di testo. Ad esempio, il comando riportato di seguito esegue una sequenza di comandi contenuti nel file di testo `myprog.in`:

```
runmqsc QMgrName < myprog.in
```

- L'operatore **>** indirizza l'emissione a un file di testo. Ad esempio, il comando riportato di seguito esegue una sequenza di comandi contenuti nel file di testo `myprog.in` e ne esegue l'output in un file denominato `results.out`:

```
runmqsc QMgrName < myprog.in > results.out
```

[Figura 1 a pagina 20](#) mostra un'estrazione dal file di comandi di MQSC `myprog.in` e [Figura 2 a pagina 20](#) mostra l'estrazione corrispondente dell'output in `results.out`.

## Esempio

I comandi MQSC sono scritti in formato leggibile, ossia in testo ASCII. Il seguente esempio è un estratto da un file di comandi di MQSC che mostra un comando MQSC, **DEFINE QLOCAL**.

```

DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;

```

Figura 1. Estrai da un file di comandi MQSC

Quando il comando **runmqsc** viene completato, viene restituito un report. Il seguente esempio è un estratto da un report:

```

Starting MQSC for queue manager jupiter.queue.manager.
.
.
12:  DEFINE QLOCAL('ORANGE.LOCAL.QUEUE') REPLACE +
:    DESCR(' ') +
:    PUT(ENABLED) +
:    DEFPRTY(0) +
:    DEFPSIST(NO) +
:    GET(ENABLED) +
:    MAXDEPTH(5000) +
:    MAXMSGL(1024) +
:    DEFSOPT(SHARED) +
:    NOHARDENBO +
:    USAGE(NORMAL) +
:    NOTRIGGER;
AMQ8006: IBM MQ queue created.
.
.
.

```

Figura 2. Estrai da un file di report di comandi MQSC

### Attività correlate

“Impostazione del prompt dei comandi MQSC” a pagina 15

È possibile impostare il prompt dei comandi MQSC su un prompt a scelta utilizzando la variabile di ambiente **MQPROMPT**.

“Esecuzione interattiva dei comandi MQSC” a pagina 13

È possibile utilizzare i comandi MQSC in modo interattivo utilizzando una finestra di comandi o una shell.

### Riferimenti correlati

[runmqsc](#)

**z/OS**

## Origini da cui è possibile emettere comandi MQSC su z/OS

I comandi MQSC possono essere emessi da varie origini, a seconda del comando.

I comandi possono essere emessi dalle seguenti origini:

- La console z/OS o equivalente
- I dataset di input di inizializzazione CSQINP1, CSQINP2, CSQINPT e CSQINPX
- Programma di utilità batch CSQUTIL
- Applicazioni autorizzate in modo appropriato, inviando comandi come messaggi al SISTEMA SYSTEM.COMMAND.INPUT

Tuttavia, non tutti i comandi possono essere emessi da tutte queste origini. I comandi possono essere classificati in base al fatto che possano essere emessi da:

**1**

CSQINP1

**2**

CSQINP2

**C**

La console z/OS

**R**

Il server dei comandi e la coda dei comandi, mediante CSQUTIL, CSQINPT, CSQINPX o applicazioni

All'interno delle descrizioni dei comandi nei [comandi MQSC](#), queste origini sono identificate dall'utilizzo dei caratteri 1, 2, C e R in ogni descrizione dei comandi.

## Configurazione automatica da uno script MQSC all'avvio

Da IBM MQ 9.2.0, è possibile configurare il gestore code per applicare automaticamente il contenuto di uno script MQSC o di una serie di script MQSC ad ogni avvio del gestore code.

È possibile utilizzare questa funzionalità per avere una configurazione che può essere modificata e riprodotta automaticamente al successivo riavvio del gestore code. Ad esempio, se lo script o gli script si trovano su un'unità montata, è possibile disporre di una configurazione centralizzata in cui l'ultima versione viene applicata a ogni gestore code all'avvio.

Uno scenario particolare in cui ciò può essere utile è garantire che un cluster uniforme contenga le stesse definizioni su tutti i gestori code nel cluster, disponendo di una singola serie di configurazione che vengono applicate tutte. Per un esempio, consultare [Creazione di un nuovo cluster uniforme](#).

### Prima di iniziare

È possibile utilizzare:

1. Un singolo script e creare un file di testo utilizzando i comandi MQSC.
2. Una serie di script MQSC:
  - Per identificare una directory in cui saranno presenti le configurazioni e
  - In tale indirizzario, creare i file, ognuno con estensione `.mqsc`, ad esempio `queues.mqsc`.

Dal momento che questo script viene riapplicato ad ogni avvio del gestore code, è importante che i comandi possano essere riprodotti. Ad esempio, un comando **DEFINE** deve contenere la stringa **REPLACE**, altrimenti il comando viene visualizzato come un errore al secondo avvio del gestore code, poiché l'oggetto esiste già.

Si noti che in uno script MQSC, qualsiasi riga con prefisso `*` viene considerata come un commento.

### Abilitazione della configurazione automatica degli script MQSC

È possibile configurare un nuovo gestore code utilizzando l'indicatore **-ic** per il comando `crtmqm` e puntando a un file specifico o a una directory. Il valore fornito viene memorizzato nel file `qm.ini` nella stanza AutoConfig, come attributo **MQSCConfig**.

È possibile configurare un gestore code esistente per abilitare la configurazione MQSC automatica, aggiungendo l'attributo della stanza AutoConfig **MQSCConfig**, che punta a un file o a una directory validi. Ad esempio:

```
AutoConfig:
MQSCConfig=C:\mq_configuration\uniclus.mqsc
```

## Come funziona la configurazione automatica?

Durante l'avvio del gestore code, la configurazione identificata dall'attributo della stanza AutoConfig **MQSCConfig** viene passata attraverso la convalida **runmqsc**, per garantire una sintassi valida e quindi memorizzata nella struttura ad albero dei dati del gestore code nella directory `autocfg` come un singolo file `cached.mqsc`.

Quando vengono elaborati più file da una directory, vengono elaborati in ordine alfabetico e se contiene un comando di fine o di uscita MQSC, il resto del contenuto di tale file viene ignorato.

Durante il primo avvio del gestore code, l'impossibilità di leggere il file o la directory o un file con sintassi MQSC non valida impedisce l'avvio del gestore code, con un messaggio di errore appropriato alla console e al log degli errori del gestore code.

Al successivo riavvio, se il file o la directory a cui si fa riferimento non è leggibile o contiene una sintassi MQSC non valida, viene utilizzato il file precedentemente memorizzato nella cache e viene evidenziato da un messaggio scritto nel log degli errori del gestore code.

**V 9.2.2** Al punto in cui il contenuto di `cached.mqsc` viene applicato al gestore code, una volta applicati tutti i comandi MQSC, il gestore code è abilitato per la connessione delle applicazioni. Il log **runmqsc** della configurazione applicata viene memorizzato nella directory degli errori del gestore code, come un file denominato `autocfgmqsc.LOG`.

Inoltre, qualsiasi comando MQSC che non viene completato correttamente, viene registrato nel log degli errori del gestore code, identificando il motivo per cui il comando ha esito negativo.

## Valori generici e caratteri con significati speciali

Alcuni caratteri, ad esempio, barra retroversa (\) e doppi apici (") I caratteri hanno significati speciali quando vengono utilizzati con i comandi MQSC. Alcuni caratteri speciali possono essere utilizzati con parametri che possono avere valori generici ma devono essere specificati correttamente.

Precedere la barra retroversa (\) e le virgolette (") con un \, ovvero, immettere \\ o \" se si desidera \ o " nel testo.

Ogni volta che un parametro può avere un valore generico, viene immesso terminando con un asterisco (\*), ad esempio ABC\*. Un valore generico indica tutti i valori che iniziano con; quindi ABC\* indica tutti i valori che iniziano con ABC. Se nel valore vengono utilizzati caratteri che richiedono le virgolette, l'asterisco deve essere inserito all'interno delle virgolette, quindi 'abc\*'. L'asterisco deve essere l'ultimo o l'unico carattere nel valore.

Il punto interrogativo (?) e i due punti (:) non sono consentiti nei valori generici.

Quando è necessario utilizzare uno di questi caratteri speciali in un campo (ad esempio, come parte di una descrizione), è necessario racchiudere l'intera stringa tra virgolette singole.

Carattere	Descrizione
	Gli spazi vengono utilizzati come separatori. Più spazi sono equivalenti ad un singolo spazio, tranne che nelle stringhe racchiuse tra apici ('). Gli eventuali spazi vuoti finali in tali attributi stringa basati sui tipi MQCHARV vengono considerati significativi.
,	Le virgole vengono utilizzate come separatori. Più virgole sono equivalenti a una singola virgola, tranne che in stringhe racchiuse tra apici (').
'	Un apostrofo indica l'inizio o la fine di una stringa. IBM MQ lascia tutti i caratteri racchiusi tra virgolette esattamente come sono stati immessi. Gli apostrofi che contengono non vengono inclusi nel calcolo della lunghezza della stringa.
"	Le virgolette singole all'interno di una stringa vengono considerate da IBM MQ come un carattere quando si calcola la lunghezza della stringa e la stringa non viene terminata.

Tabella 1. Descrizioni di caratteri con significati speciali (Continua)

Carattere	Descrizione
=	 Su z/OS, un segno di uguale indica l'inizio di un valore di parametro che termina con una virgola o uno spazio.
(	Una parentesi aperta indica l'inizio di un valore di parametro o di un elenco di valori.
)	Una parentesi di chiusura indica la fine di un valore di parametro o di un elenco di valori.
:	I due punti indicano un intervallo inclusivo. Ad esempio (1: 5) indica (1,2,3,4, 5). Questa notazione può essere utilizzata solo nei comandi <b>TRACE</b> .
*	Un asterisco indica tutto. Ad esempio, <b>DISPLAY TRACE (*)</b> indica di visualizzare tutte le tracce e <b>DISPLAY QUEUE (PAY*)</b> indica di visualizzare tutte le code con nomi che iniziano con PAY.

## Automazione della gestione IBM MQ utilizzando i comandi PCF

Si potrebbe decidere che sarebbe utile per l'installazione automatizzare alcune attività di gestione e monitoraggio. È possibile automatizzare le attività di gestione per i gestori code locali e remoti utilizzando i comandi PCF (Programmable Command Format). Questa sezione presuppone che si abbia esperienza nella gestione degli oggetti IBM MQ .

### Comandi PCF

I comandi PCF (Programmable Command Format) IBM MQ possono essere utilizzati per programmare le attività di gestione in un programma di gestione. In tal modo, da un programma è possibile manipolare gli oggetti gestore code (code, definizioni di processi, elenchi nomi, canali, canali di connessione client, listener, servizi e oggetti delle informazioni di autenticazione) e persino gestire i gestori code stessi.

I comandi PCF coprono la stessa gamma delle funzioni fornite dai comandi MQSC. È possibile scrivere un programma per emettere comandi PCF per qualsiasi gestore code nella rete da un singolo nodo. In questo modo, è possibile centralizzare e automatizzare le attività di gestione.

Ogni comando PCF è una struttura di dati integrata nella parte di dati dell'applicazione di un messaggio IBM MQ . Ogni comando viene inviato al gestore code di destinazione utilizzando la funzione MQI MQPUT allo stesso modo di qualsiasi altro messaggio. Se il server dei comandi è in esecuzione sul gestore code che riceve il messaggio, il server dei comandi lo interpreta come un messaggio di comando ed esegue il comando. Per ottenere le risposte, l'applicazione emette una chiamata MQGET e i dati della risposta vengono restituiti in un'altra struttura dati. La domanda può quindi elaborare la risposta e agire di conseguenza.

**Nota:** Diversamente dai comandi MQSC, i comandi PCF e le relative risposte non sono in un formato di testo che è possibile leggere.

Brevemente, queste sono alcune delle cose necessarie per creare un messaggio di comando PCF:

#### Descrittore messaggio

Questo è un descrittore di messaggi IBM MQ standard, in cui:

- Il tipo di messaggio (*MsqType*) è MQMT\_REQUEST.
- Il formato del messaggio (*Format*) è MQFMT\_ADMIN.

#### Dati applicazione

Contiene il messaggio PCF incluso l'intestazione PCF, in cui:

- Il tipo di messaggio PCF (*Type*) specifica MQCFT\_COMMAND.
- L'identificativo del comando specifica il comando, ad esempio *Change Queue* (MQCMD\_CHANGE\_Q).

Per una descrizione completa delle strutture di dati PCF e come implementarle, consultare [“Introduzione a IBM MQ Programmable Command Formats”](#) a pagina 24.

## Attributi oggetto PCF

Gli attributi dell'oggetto in PCF non sono limitati a otto caratteri come per i comandi MQSC. Sono mostrati in questa guida in corsivo. Ad esempio, l'equivalente PCF di RQMNAME è *RemoteQMGrName*.

## PCF di escape

I PCF di escape sono comandi PCF che contengono comandi MQSC all'interno del testo del messaggio. È possibile utilizzare i PCF per inviare comandi a un gestore code remoto. Per ulteriori informazioni sui PCF di escape, consultare [Escape](#).

## Introduzione a IBM MQ Programmable Command Formats

I PCF (Programmable Command Format) definiscono i messaggi di comando e risposta che possono essere scambiati tra un programma e qualsiasi gestore code (che supporta i PCF) in una rete. I PCF semplificano la gestione dei gestori code e di altre reti. Possono essere utilizzati per risolvere il problema della gestione complessa di reti distribuite, soprattutto quando le reti crescono in dimensioni e complessità.

I formati dei comandi programmabili sono supportati da:

-  IBM MQ for AIX
-  IBM MQ for IBM i
-  IBM MQ per Linux
-  IBM MQ for Windows
-  IBM MQ for z/OS

## Il problema che i comandi PCF risolvono

La gestione delle reti distribuite può diventare complessa. I problemi amministrativi continuano a crescere con l'aumento delle dimensioni e della complessità delle reti.

Esempi di gestione specifica per la messaggistica e l'accodamento includono:

- Gestione delle risorse.  
Ad esempio, creazione ed eliminazione della coda.
- Monitoraggio delle prestazioni.  
Ad esempio, la profondità massima della coda o la frequenza dei messaggi.
- Controllo.  
Ad esempio, l'ottimizzazione dei parametri della coda come la profondità massima della coda, la lunghezza massima del messaggio e l'abilitazione e la disabilitazione delle code.
- Intradamento del messaggio.  
Definizione di percorsi alternativi attraverso una rete.

I comandi IBM MQ PCF possono essere utilizzati per semplificare la gestione del gestore code e di altre reti. I comandi PCF consentono di utilizzare una singola applicazione per eseguire la gestione della rete da un singolo gestore code nella rete.

## Che cosa sono i PCF?

I PCF definiscono i messaggi di comando e di risposta che possono essere scambiati tra un programma e qualsiasi gestore code (che supporta PCF) in una rete. È possibile utilizzare i comandi PCF in un programma di applicazione di gestione dei sistemi per la gestione degli oggetti IBM MQ : oggetti delle informazioni di autenticazione, canali, listener dei canali, elenchi nomi, definizioni di processi, gestori code, code, servizi e classi di memoria. L'applicazione può operare da un singolo punto nella rete per comunicare le informazioni di comando e risposta con qualsiasi gestore code, locale o remoto, utilizzando il gestore code locale.

Ciascun gestore code ha una coda di gestione con un nome coda standard e l'applicazione può inviare messaggi di comando PCF a tale coda. Ogni gestore code ha anche un server dei comandi per gestire i messaggi di comando dalla coda di amministrazione. I messaggi di comando PCF possono quindi essere elaborati da qualsiasi gestore code nella rete e i dati di risposta possono essere restituiti all'applicazione, utilizzando la coda di risposta specificata. I comandi PCF e i messaggi di risposta vengono inviati e ricevuti utilizzando la normale MQI (Message Queue Interface).

Per un elenco dei comandi PCF disponibili, inclusi i relativi parametri, consultare [Definizioni dei formati dei comandi programmabili](#).

## Utilizzo di IBM MQ Programmable Command Format

È possibile utilizzare i PCF in un programma di gestione dei sistemi per la gestione remota IBM MQ .

Questa sezione include:

- [“Messaggi di comando PCF” a pagina 25](#)
- [“Risposte PCF in IBM MQ” a pagina 28](#)
-  [“Risposte estese” a pagina 30](#)
- [Regole per la denominazione di oggetti IBM MQ](#)
- [“Controllo autorizzazione per comandi PCF in IBM MQ” a pagina 31](#)

### Messaggi di comando PCF

I messaggi di comando PCF sono costituiti da un'intestazione PCF, parametri identificati in tale intestazione e anche dati di messaggi definiti dall'utente. I messaggi vengono emessi utilizzando le chiamate dell'interfaccia della coda messaggi.

Ogni comando e i relativi parametri vengono inviati come un messaggio di comando separato contenente un'intestazione PCF seguita da un numero di strutture di parametri; per i dettagli dell'intestazione PCF, consultare [MQCFH - PCF header](#) per un esempio di una struttura di parametri, consultare [MQCFST - PCF string parameter](#). L'intestazione PCF identifica il comando e il numero di strutture di parametri che seguono nello stesso messaggio. Ogni struttura di parametri fornisce un parametro al comando.

Le risposte ai comandi, generate dal server dei comandi, hanno una struttura simile. C'è un'intestazione PCF, seguita da un numero di strutture di parametri. Le risposte possono essere costituite da più di un messaggio, ma i comandi sono sempre composti da un solo messaggio.

 Su [Multiplatforms](#), la coda a cui vengono inviati i comandi PCF viene sempre denominata `SYSTEM.ADMIN.COMMAND.QUEUE`.

 Su [z/OS](#), i comandi vengono inviati a `SYSTEM.COMMAND.INPUT`, sebbene `SYSTEM.ADMIN.COMMAND.QUEUE` può essere un alias per esso. Il server dei comandi che serve questa coda invia le risposte alla coda definita dai campi `ReplyToQ` e `ReplyToQMGr` nel descrittore del messaggio del messaggio di comando.

### Come emettere i messaggi di comando PCF

Utilizzare le normali chiamate MQI (Message Queue Interface), `MQPUT`, `MQGET` e così via, per inserire e richiamare il comando PCF e i messaggi di risposta da e verso le relative code.

**Nota:**

Assicurarsi che il server dei comandi sia in esecuzione sul gestore code di destinazione per il comando PCF da elaborare su tale gestore code.

Per un elenco dei file di intestazione forniti, consultare [IBM MQ COPY, header, include e module files](#).

**Descrittore del messaggio per un comando PCF**

Il descrittore del messaggio IBM MQ è completamente documentato in [MQMD - Descrittore del messaggio](#).

Un messaggio di comando PCF contiene i seguenti campi nel descrittore del messaggio:

**Prospetto**

Qualsiasi valore valido, come richiesto.

**MsgType**

Questo campo deve essere MQMT\_REQUEST per indicare un messaggio che richiede una risposta.

**Scadenza**

Qualsiasi valore valido, come richiesto.

**Feedback**

Imposta su MQFB\_NONE

**Multi Codifica**

Se si sta inviando a un sistema IBM MQ for Multiplatforms , impostare questo campo sulla codifica utilizzata per i dati del messaggio. La conversione viene eseguita, se necessario.

**Multi CodedCharSetId**

Se si sta inviando a un sistema IBM MQ for Multiplatforms , impostare questo campo sull'identificativo della serie di caratteri codificati utilizzato per i dati del messaggio. La conversione viene eseguita, se necessario.

**Formato**

Impostare su MQFMT\_ADMIN.

**Priorit...**

Qualsiasi valore valido, come richiesto.

**Persistenza**

Qualsiasi valore valido, come richiesto.

**MsgId**

L'applicazione di invio può specificare qualsiasi valore oppure è possibile specificare MQMI\_NONE per richiedere al gestore code di creare un identificativo di messaggio univoco.

**CorrelId**

L'applicazione mittente può specificare qualsiasi valore oppure MQCI\_NONE può essere specificato per indicare nessun identificativo di correlazione.

**ReplyToQ**

Il nome della coda per ricevere la risposta.

**ReplyToQMgr**

Il nome del gestore code per la risposta (o vuoto).

**Campi di contesto del messaggio**

Questi campi possono essere impostati su qualsiasi valore valido, come richiesto. Di solito, l'opzione di inserimento del messaggio MQPMO\_DEFAULT\_CONTEXT viene utilizzata per impostare i campi di contesto del messaggio sui valori predefiniti.

Se si utilizza una struttura MQMD version-2 , è necessario impostare i seguenti campi aggiuntivi:

**GroupId**

Imposta su MQGI\_NONE

**MsgSeqNumber**

Imposta su 1

**Offset**

Imposta su 0

**MsgFlags**

Imposta su MQMF\_NONE

**OriginalLength**

Imposta su MQOL\_UNDEFINED

**Invio di dati utente**

Le strutture PCF possono essere utilizzate anche per inviare dati di messaggi definiti dall'utente. In questo caso, il campo *Format* del descrittore messaggi deve essere impostato su MQFMT\_PCF.

**Invio e ricezione di messaggi PCF in una coda specificata****Invio di messaggi PCF a una coda specificata**

Per inviare un messaggio a una coda specificata, la chiamata mqPutBag converte il contenuto della borsa specificata in un messaggio PCF e invia il messaggio alla coda specificata. Il contenuto della borsa rimane invariato dopo la chiamata.

Come input per questa chiamata, è necessario fornire:

- Un handle di connessione MQI.
- Un handle dell'oggetto per la coda su cui deve essere inserito il messaggio.
- Un descrizione del messaggio. Per ulteriori informazioni sul descrittore del messaggio, consultare [MQMD - Descrittore del messaggio](#).
- Inserire le opzioni del messaggio utilizzando la struttura MQPMO. Per ulteriori informazioni sulla struttura MQPMO, consultare [MQPMO - Put - message options](#).
- La maniglia del contenitore da convertire in un messaggio.

**Nota:** Se il contenitore contiene un messaggio di gestione e la chiamata mqAddInquiry è stata utilizzata per inserire i valori nel contenitore, il valore dell'elemento di dati MQIASY\_COMMAND deve essere un comando INQUIRE riconosciuto da MQAI.

Per una descrizione completa della chiamata mqPutBag, consultare [mqPutBag](#).

**Ricezione di messaggi PCF da una coda specificata**

Per ricevere un messaggio da una coda specificata, la chiamata mqGetBag richiama un messaggio PCF da una coda specificata e converte i dati del messaggio in una serie di dati.

Come input per questa chiamata, è necessario fornire:

- Un handle di connessione MQI.
- Una gestione oggetto della coda da cui deve essere letto il messaggio.
- Un descrizione del messaggio. Nella struttura MQMD, il parametro **Format** deve essere MQFMT\_ADMIN, MQFMT\_EVENT o MQFMT\_PCF.

**Nota:** Se il messaggio viene ricevuto all'interno di un'unità di lavoro (con l'opzione MQGMO\_SYNCPOINT) e il messaggio ha un formato non supportato, è possibile eseguire il backout dell'unità di lavoro. Il messaggio viene quindi reintegrato sulla coda e può essere recuperato utilizzando la chiamata MQGET invece della chiamata mqGetBag. Per ulteriori informazioni sul descrittore del messaggio, consultare [MQGMO - Get - message options](#).

- Richiamare le opzioni del messaggio utilizzando la struttura MQGMO. Per ulteriori informazioni sulla struttura MQGMO, consultare [MQMD - Message Descriptor](#).
- L'handle del contenitore per contenere il messaggio convertito.

Per una descrizione completa della chiamata mqGetBag, vedere [mqGetBag](#).

### **Risposte PCF in IBM MQ**

In risposta a ciascun comando, il server dei comandi genera uno o più messaggi di risposta. Un messaggio di risposta ha un formato simile a un messaggio di comando.

L'intestazione PCF ha lo stesso valore identificativo del comando a cui è una risposta (consultare [MQCFH - intestazione PCF](#) per i dettagli). L'identificativo del messaggio e l'identificativo di correlazione vengono impostati in base alle opzioni di report della richiesta.

Se il tipo di intestazione PCF del messaggio di comando è MQCFT\_COMMAND, vengono generate solo le risposte standard. Tali comandi sono supportati su tutte le piattaforme tranne z/OS. Le applicazioni meno recenti non supportano PCF su z/OS ; IBM MQ Windows Explorer è una di tali applicazioni (tuttavia, IBM MQ Explorer IBM WebSphere MQ 6.0 o versioni successive supporta PCF su z/OS ).

Se il tipo di intestazione PCF del messaggio di comando è MQCFT\_COMMAND\_XR, vengono generate risposte estese o standard. Tali comandi sono supportati su z/OS e su alcune altre piattaforme. I comandi emessi su z/OS generano solo risposte estese. Su altre piattaforme, è possibile che venga generato un tipo di risposta.

Se un singolo comando specifica un nome oggetto generico, viene restituita una risposta separata nel proprio messaggio per ogni oggetto corrispondente. Per la generazione della risposta, un singolo comando con un nome generico viene considerato come più comandi singoli (tranne che per il campo di controllo MQCFC\_LAST o MQCFC\_NOT\_LAST). In caso contrario, un messaggio di comando genera un messaggio di risposta.

Alcune risposte PCF potrebbero restituire una struttura anche quando non è richiesta. Questa struttura viene mostrata nella definizione della risposta ( [Definizioni dei formati di comando programmabili](#) ) come *sempre restituito*. Il motivo per cui, per queste risposte, è necessario denominare gli oggetti nella risposta per identificare quale oggetto vengono applicati i dati.

### **Descrittori di messaggi per una risposta**

Un messaggio di risposta ha i seguenti campi nel descrittore del messaggio:

#### **MsgType**

Questo campo è MQMT\_REPLY.

#### **MsgId**

Questo campo viene generato dal gestore code.

#### **CorrelId**

Questo campo viene generato in base alle opzioni di report del messaggio di comando.

#### **Formato**

Questo campo è MQFMT\_ADMIN.

#### **Codifica**

Impostare su MQENC\_NATIVE.

#### **CodedCharSetId**

Impostare su MQCCSI\_Q\_MGR.

#### **Persistenza**

Lo stesso come nel messaggio di comando.

#### **Priorit...**

Lo stesso come nel messaggio di comando.

La risposta viene generata con MQPMO\_PASS\_IDENTITY\_CONTEXT.

### *Risposte standard*

Messaggi di comando con un tipo di intestazione MQCFT\_COMMAND, vengono generate risposte standard. Tali comandi sono supportati su tutte le piattaforme tranne z/OS.

Esistono tre tipi di risposta standard:

- Risposta OK
- Risposta di errore
- Risposta dati

### **Risposta OK**

Questa risposta è costituita da un messaggio che inizia con un'intestazione del formato del comando, con un campo *CompCode* MQCC\_OK o MQCC\_WARNING.

Per MQCC\_OK, *Reason* è MQRC\_NONE.

Per MQCC\_WARNING, *Reason* identifica la natura dell'avvertenza. In questo caso, l'intestazione del formato del comando potrebbe essere seguita da una o più strutture di parametri di avvertenza appropriate a questo codice di errore.

In entrambi i casi, per un comando inquire potrebbero seguire ulteriori strutture di parametri, come descritto nelle seguenti sezioni.

### **Risposta di errore**

Se il comando ha un errore, vengono inviati uno o più messaggi di risposta di errore (più di uno potrebbe essere inviato anche per un comando che normalmente avrebbe un solo messaggio di risposta). Questi messaggi di risposta di errore hanno MQCFC\_LAST o MQCFC\_NOT\_LAST impostati come appropriato.

Ogni messaggio inizia con un'intestazione del formato della risposta, con un valore *CompCode* MQCC\_FAILED e un campo *Reason* che identifica il particolare errore. In generale, ogni messaggio descrive un errore differente. Inoltre, ogni messaggio ha zero o una (mai più di una) struttura di parametri di errore dopo l'intestazione. Questa struttura di parametri, se presente, è una struttura MQCFIN, con un campo *Parameter* che contiene uno dei seguenti:

- ID\_PARAMETER\_MQIACF

Il campo *Value* nella struttura è l'identificativo del parametro in errore (ad esempio, MQCA\_Q\_NAME).

- ID\_ERRORE\_MQIACF

Questo valore viene utilizzato con un valore *Reason* (nell'intestazione del formato del comando) di MQRC\_UNEXPECTED\_ERROR. Il campo *Value* nella struttura MQCFIN rappresenta il codice motivo non previsto ricevuto dal server dei comandi.

- MQIACF\_SELECTOR

Questo valore si verifica se una struttura di elenco (MQCFIL) inviata con il comando contiene un selettore duplicato o uno non valido. Il campo *Reason* nell'intestazione del formato del comando identifica l'errore e il campo *Value* nella struttura MQCFIN è il valore del parametro nella struttura MQCFIL del comando che era in errore.

- ERRORE\_MQIACF\_OFFSET

Questo valore si verifica quando si verifica un errore di confronto dati nel comando Canale di ping. Il campo *Value* nella struttura è l'offset dell'errore di confronto del Canale di ping.

- ID\_MQIA\_CODED\_CHAR\_SET\_

Questo valore si verifica quando il CCSID (coded character set identifier) nel descrittore del messaggio del comando PCF in entrata non corrisponde a quello del gestore code di destinazione. Il campo *Value* nella struttura è l'identificativo della serie di caratteri codificati del gestore code.

L'ultimo (o unico) messaggio di risposta di errore è una risposta di riepilogo, con un campo *CompCode* di MQCC\_FAILED e un campo *Reason* di MQRCCF\_COMMAND\_FAILED. Questo messaggio non ha una struttura di parametri che segue l'intestazione.

## Risposta dati

Questa risposta consiste in una risposta OK (come descritto in precedenza) a un comando inquire. La risposta OK è seguita da strutture aggiuntive contenenti i dati richiesti come descritto in [Definizioni dei formati di comando programmabili](#).

Le applicazioni non devono dipendere da queste strutture di parametri aggiuntive che vengono restituite in un ordine particolare.

### Risposte estese

I comandi emessi su z/OS generano risposte estese.

Esistono tre tipi di risposta estesa:

- Risposta del messaggio, con tipo MQCFT\_XR\_MSG
- Risposta elemento, con tipo MQCFT\_XR\_ITEM
- Risposta di riepilogo, con tipo MQCFT\_XR\_SUMMARY

Ogni comando può generare una o più serie di risposte. Ogni serie di risposte comprende uno o più messaggi, numerati in sequenza da 1 nel campo *MsgSeqNumber* dell'intestazione PCF. Il campo *Control* dell'ultima (o unica) risposta in ogni serie ha il valore MQCFC\_LAST. Per tutte le altre risposte nell'insieme, questo valore è MQCFC\_NOT\_LAST.

Qualsiasi risposta può includere una o più strutture MQCFBS facoltative in cui il campo *Parameter* è impostato su MQBACF\_RESPONSE\_SET, il valore è un identificativo del set di risposte. Gli identificatori sono univoci e identificano la serie di risposte che contengono la risposta. Per ogni serie di risposte, esiste una struttura MQCFBS che la identifica.

Le risposte estese hanno almeno due strutture di parametro:

- Una struttura MQCFBS con campo *Parameter* impostato su MQBACF\_RESPONSE\_ID. Il valore in questo campo è l'identificatore della serie di risposte a cui appartiene la risposta. L'identificativo nella prima serie è arbitrario. Nelle serie successive, l'identificativo è quello precedentemente notificato in una struttura MQBACF\_RESPONSE\_SET.
- Una struttura MQCFST con il campo *Parameter* impostato su MQCACF\_RESPONSE\_Q\_MGR\_NAME, il valore è il nome del gestore code da cui proviene la serie di risposte.

Molte risposte hanno strutture di parametri aggiuntive e queste strutture sono descritte nelle sezioni seguenti.

Non è possibile determinare in anticipo il numero di risposte presenti in una serie se non ottenendo risposte fino a quando non ne viene trovata una con MQCFC\_LAST. Inoltre, non è possibile determinare in anticipo quanti insiemi di risposte ci sono, poiché qualsiasi insieme potrebbe includere strutture MQBACF\_RESPONSE\_SET per indicare che vengono generati ulteriori insiemi.

## Risposte estese ai comandi di interrogazione

I comandi inquire generano normalmente una risposta elemento (tipo MQCFT\_XR\_ITEM) per ogni elemento trovato che corrisponde ai criteri di ricerca specificati. La risposta elemento ha un campo *CompCode* nell'intestazione con un valore di MQCC\_OK e un campo *Reason* con un valore di MQRC\_NONE. Include anche altre strutture di parametri che descrivono l'elemento e gli attributi richiesti, come descritto in [Definizioni dei formati di comando programmabili](#).

Se un elemento è in errore, il campo *CompCode* nell'intestazione ha un valore di MQCC\_FAILED e il campo *Reason* identifica il particolare errore. Sono incluse ulteriori strutture di parametri per identificare l'elemento.

Alcuni comandi Inquire potrebbero restituire risposte di messaggi generali (non specifiche del nome) in aggiunta alle risposte dell'elemento. Queste risposte sono informazioni o errori di tipo MQCFT\_XR\_MSG.

Se il comando Inquire riesce, potrebbe, facoltativamente, essere una risposta di riepilogo (tipo MQCFT\_XR\_SUMMARY), con un valore *CompCode* di MQCC\_OK e un valore del campo *Reason* di MQRC\_NONE.

Se il comando Inquire ha esito negativo, potrebbero essere restituite risposte di elementi e, facoltativamente, potrebbe essere presente una risposta di riepilogo (tipo MQCFT\_XR\_SUMMARY), con un valore *CompCode* di MQCC\_FAILED e un valore del campo *Reason* di MQRCCF\_COMMAND\_FAILED.

## Risposte estese ai comandi diversi da Inquire

I comandi riusciti generano risposte ai messaggi in cui il campo *CompCode* nell'intestazione ha un valore di MQCC\_OK e il campo *Reason* ha un valore di MQRC\_NONE. C'è sempre almeno un messaggio; potrebbe essere informativo (MQCFT\_XR\_MSG) o un riepilogo (MQCFT\_XR\_SUMMARY). Potrebbero essere presenti ulteriori messaggi informativi (tipo MQCFT\_XR\_MSG). Ogni messaggio informativo potrebbe includere un certo numero di strutture di parametri aggiuntive con informazioni sul comando; consultare le descrizioni dei singoli comandi per le strutture che possono verificarsi.

I comandi che hanno esito negativo generano risposte di messaggio di errore (tipo MQCFT\_XR\_MSG), in cui il campo *CompCode* nell'intestazione ha un valore MQCC\_FAILED e il campo *Reason* identifica il particolare errore. Ogni messaggio potrebbe includere un numero di strutture di parametri aggiuntive con informazioni sull'errore: consultare le descrizioni degli errori individuali per le strutture che possono verificarsi. Potrebbero essere generate risposte di messaggi informativi. Potrebbe, facoltativamente, essere presente una risposta di riepilogo (MQCFT\_XR\_SUMMARY), con un valore *CompCode* di MQCC\_FAILED e un valore del campo *Reason* di MQRCCF\_COMMAND\_FAILED.

## Risposte estese ai comandi utilizzando CommandScope

Se un comando utilizza il parametro **CommandScope** o fa in modo che venga generato un comando utilizzando il parametro **CommandScope**, c'è un set di risposte iniziale dal gestore code in cui è stato ricevuto il comando. Quindi, viene generata una serie separata di risposte per ogni gestore code a cui viene indirizzato il comando (come se fossero stati emessi più comandi singoli). Infine, esiste una serie di risposte dal gestore code di ricezione che include una risposta di riepilogo generale (tipo MQCFT\_XR\_SUMMARY). La struttura del parametro MQCACF\_RESPONSE\_Q\_MGR\_NAME identifica il gestore code che genera ciascun set.

La serie di risposte iniziale ha le seguenti strutture di parametri aggiuntivi:

- MQIACF\_COMMAND\_INFO (MQCFIN). I valori possibili in questa struttura sono MQCMDI\_CMDSCOPE\_ACCEPTED o MQCMDI\_CMDSCOPE\_GENERATED.
- MQIACF\_CMDSCOPE\_Q\_MGR\_COUNT (MQCFIN). Questa struttura indica il numero di gestori code a cui viene inviato il comando.

## Controllo autorizzazione per comandi PCF in IBM MQ

Quando viene elaborato un comando PCF, il *UserIdentifier* dal descrittore del messaggio nel messaggio di comando viene utilizzato per i controlli di autorizzazione dell'oggetto IBM MQ richiesti. Il controllo delle autorizzazioni viene implementato in maniera diversa su ciascuna piattaforma, come descritto in questo argomento.

I controlli vengono eseguiti sul sistema su cui viene elaborato il comando; pertanto, questo ID utente deve esistere sul sistema di destinazione e disporre delle autorizzazioni richieste per elaborare il comando. Se il messaggio proviene da un sistema remoto, un modo per ottenere l'ID esistente sul sistema di destinazione è avere un ID utente corrispondente sia sul sistema locale che su quello remoto.

**Nota:**  Per informazioni sul controllo delle autorizzazioni su z/OS, consultare [Attività 1: Identificazione dei parametri di sistema z/OS](#).

## IBM MQ for IBM i

IBM i

Per elaborare qualsiasi comando PCF, l'ID utente deve disporre dell'autorizzazione *dsp* per l'oggetto IBM MQ sul sistema di destinazione.

Inoltre, i controlli dell'autorizzazione dell'oggetto IBM MQ vengono eseguiti per alcuni comandi PCF, come mostrato in [Tabella 2 a pagina 33](#).

Nella maggior parte dei casi questi controlli sono gli stessi dei controlli eseguiti dai comandi CL IBM MQ equivalenti emessi su un sistema locale. Consultare Impostazione della sicurezza su IBM i, per ulteriori informazioni sulla corrispondenza tra le autorizzazioni IBM MQ e le autorizzazioni di sistema IBM i e i requisiti di autorizzazione per i comandi CL IBM MQ. I dettagli sulla sicurezza relativi alle uscite sono forniti nella documentazione [Sicurezza a livello di collegamento utilizzando un'uscita di sicurezza](#).

**Per elaborare uno dei seguenti comandi** l'ID utente deve essere un membro del profilo di gruppo QMQMADM:

- Ping canale
- Modifica canale
- Copia canale
- Creazione canale
- Eliminazione canale
- Reimposta canale
- Risolvi canale
- Avvio canale
- Arresta canale
- Avvio iniziatore canale
- Avvio listener canale

## IBM MQ for UNIX, Linux, and Windows

ALW

Per elaborare qualsiasi comando PCF, l'ID utente deve disporre dell'autorizzazione *dsp* per l'oggetto gestore code sul sistema di destinazione. Inoltre, i controlli dell'autorizzazione dell'oggetto IBM MQ vengono eseguiti per alcuni comandi PCF, come mostrato in [Tabella 2 a pagina 33](#).

**Per elaborare uno dei seguenti comandi** l'ID utente deve appartenere al gruppo *mqm*.

**Nota:** Solo per Windows, l'ID utente può appartenere al gruppo *Administrators* o al gruppo *mqm*.

- Modifica canale
- Copia canale
- Creazione canale
- Eliminazione canale
- Ping canale
- Reimposta canale
- Avvio canale
- Arresta canale
- Avvio iniziatore canale
- Avvio listener canale
- Risolvi canale
- Reimposta cluster

- Aggiornamento cluster
- Sospensione gestore code
- Ripristino gestore code

## Autorizzazioni oggetto IBM MQ per Multiplatforms



<i>Tabella 2. Autorizzazioni oggetto</i>		
<b>Comando</b>	<b>IBM MQ Autorizzazione di oggetti</b>	<b>Autorizzazione classe (per tipo di oggetto)</b>
Modifica informazioni di autenticazione	dsp e chg	n/a
Modifica canale	dsp e chg	n/a
Modifica listener canale	dsp e chg	n/a
Modifica canale di connessione client	dsp e chg	n/a
Modifica elenco nomi	dsp e chg	n/a
Modifica processo	dsp e chg	n/a
Modifica coda	dsp e chg	n/a
Modifica gestore code	chg <i>vedere Nota 3 e Nota 5</i>	n/a
Modifica servizio	dsp e chg	n/a
Cancellazione coda	clr	n/a
Copia informazioni di autenticazione	dsp	crt
Copia informazioni di autenticazione (Sostituisci) <i>vedere la Nota 1</i>	<i>da: dsp a chg</i>	crt
Copia canale	dsp	crt
Copia canale (Sostituisci) <i>vedere la nota 1</i>	<i>da: dsp a chg</i>	crt
Copia listener canale	dsp	crt
Copiare il listener del canale (Sostituisci) <i>vedere la nota 1</i>	<i>da: dsp a chg</i>	crt
Copia canale di connessione client	dsp	crt
Copia canale di connessione client (Sostituisci) <i>vedere la nota 1</i>	<i>da: dsp a chg</i>	crt
Copia elenco nomi	dsp	crt
Copia elenco nomi (Sostituisci) <i>vedere la Nota 1</i>	<i>da: dsp a dsp e chg</i>	crt
Copia processo	dsp	crt

Tabella 2. Autorizzazioni oggetto (Continua)

<b>Comando</b>	<b>IBM MQ Autorizzazione di oggetti</b>	<b>Autorizzazione classe (per tipo di oggetto)</b>
Copia processo (Sostituisci) <i>vedere Nota 1</i>	<i>da: dsp a chg</i>	crt
Copia coda	dsp	crt
Copia coda (Sostituisci) <i>vedere la nota 1</i>	<i>da: dsp a dsp e chg</i>	crt
Creazione informazioni di autenticazione	<i>(informazioni di autenticazione predefinite del sistema) dsp</i>	crt
Crea informazioni di autenticazione (Sostituisci) <i>vedere la Nota 1</i>	<i>(informazioni di autenticazione predefinite del sistema) dsp a: chg</i>	crt
Creazione canale	<i>(canale predefinito del sistema) dsp</i>	crt
Crea canale (Sostituisci) <i>vedere la nota 1</i>	<i>(canale predefinito del sistema) dsp in: chg</i>	crt
Crea listener del canale	<i>(listener predefinito di sistema) dsp</i>	crt
Crea listener canale (Sostituisci) <i>vedere la nota 1</i>	<i>(listener predefinito di sistema) dsp in: chg</i>	crt
Crea canale di connessione client	<i>(canale predefinito del sistema) dsp</i>	crt
Crea canale di connessione client (Sostituisci) <i>vedere la nota 1</i>	<i>(canale predefinito del sistema) dsp in: chg</i>	crt
Creazione elenco nomi	<i>(elenco nomi predefinito di sistema) dsp</i>	crt
Crea elenco nomi (Sostituisci) <i>vedere la Nota 1</i>	<i>(elenco nomi predefinito di sistema) dsp su: dsp e chg</i>	crt
Creazione processo	<i>(processo predefinito del sistema) dsp</i>	crt
Crea processo (Sostituisci) <i>vedere la nota 1</i>	<i>(processo predefinito di sistema) dsp to: chg</i>	crt
Creazione coda	<i>(coda predefinita di sistema) dsp</i>	crt
Crea coda (Sostituisci) <i>vedere la Nota 1</i>	<i>(coda predefinita del sistema) dsp a: dsp e chg</i>	crt
Crea servizio	<i>(coda predefinita di sistema) dsp</i>	crt
Crea servizio (Sostituisci) <i>vedere Nota 1</i>	<i>(coda predefinita di sistema) dsp per: chg</i>	crt
Eliminazione informazioni di autenticazione	dsp e dlt	n/a
Eliminare il record di autorizzazione	<i>(oggetto gestore code) chg vedere la Nota 4</i>	<i>vedere la Nota 4</i>
Eliminazione canale	dsp e dlt	n/a
Elimina listener canale	dsp e dlt	n/a
Elimina canale di connessione client	dsp e dlt	n/a

Tabella 2. Autorizzazioni oggetto (Continua)

Comando	IBM MQ Autorizzazione di oggetti	Autorizzazione classe (per tipo di oggetto)
Eliminazione elenco nomi	dsp e dlt	n/a
Eliminazione processo	dsp e dlt	n/a
Eliminazione coda	dsp e dlt	n/a
Elimina servizio	dsp e dlt	n/a
Interrogazione informazioni di autenticazione	dsp	n/a
Interrogazione record autorizzazione	<i>vedere la Nota 4</i>	<i>vedere la Nota 4</i>
Interrogazione canale	dsp	n/a
Interroga listener canale	dsp	n/a
Inquire Channel Status (per <b>ChannelType</b> MQCHT_CLSSDR)	inq	n/a
Interroga canale di connessione client	dsp	n/a
Interrogazione elenco nomi	dsp	n/a
Interrogazione processo	dsp	n/a
Interrogazione coda	dsp	n/a
Interrogazione gestore code	<i>vedere la nota 3</i>	n/a
Interrogazione stato coda	dsp	n/a
Interrogazione servizio	dsp	n/a
Ping canale	ctrl	n/a
Ping gestore code	<i>vedere la nota 3</i>	n/a
Aggiornamento gestore code	(oggetto gestore code) chg	n/a
Aggiorna sicurezza (per <b>SecurityType</b> MQSECTYPE_SSL)	(oggetto gestore code) chg	n/a
Reimposta canale	ctrlx	n/a
Reimpostazione gestore code	(oggetto gestore code) chg	n/a
Reimposta statistiche coda	dsp e chg	n/a
Risolvi canale	ctrlx	n/a
Imposta record di autorizzazione	<i>(oggetto gestore code) chg vedere la Nota 4</i>	<i>vedere la Nota 4</i>
Avvio canale	ctrl	n/a
Arresta canale	ctrl	n/a
Arresta connessione	(oggetto gestore code) chg	n/a
Avvia listener	ctrl	n/a
Arresto del listener	ctrl	n/a

Tabella 2. Autorizzazioni oggetto (Continua)		
Comando	IBM MQ Autorizzazione di oggetti	Autorizzazione classe (per tipo di oggetto)
Avvia servizio	ctrl	n/a
Arresta servizio	ctrl	n/a
Esc	vedere la Nota 2	vedere la Nota 2

**Note:**

1. Questo comando si applica se l'oggetto da sostituire esiste, altrimenti il controllo dell'autorizzazione è quello per la creazione o la copia senza sostituzione.
2. L'autorizzazione richiesta è determinata dal comando MQSC definito dal testo di escape ed è equivalente a uno dei precedenti comandi.
3. Per poter elaborare qualsiasi comando PCF, l'ID utente deve disporre dell'autorizzazione dsp per l'oggetto gestore code sul sistema di destinazione.
4. Questo comando PCF è autorizzato a meno che il server dei comandi non sia stato avviato con il parametro -a. Per impostazione predefinita, il server dei comandi viene avviato all'avvio del gestore code e senza il parametro -a. Per ulteriori informazioni, consultare [Programmable command formats reference](#).
5. La concessione dell'autorizzazione chg per un gestore code consente di impostare i record di autorizzazioni per tutti i gruppi e gli utenti. Non concedere questa autorizzazione agli utenti ordinari o alle applicazioni.

IBM MQ fornisce anche alcuni punti di uscita di sicurezza del canale in modo da poter fornire i propri programmi di uscita utente per il controllo di sicurezza. Per ulteriori informazioni, consultare [Visualizzazione di un canale](#).

**Multi**

## Utilizzo di MQAI per semplificare l'utilizzo dei PCF

MQAI ( IBM MQ Administration Interface) è un'interfaccia di programmazione per IBM MQ disponibile su AIX, IBM i, Linux, e Windows. Esegue le attività di gestione su un gestore code IBM MQ utilizzando i bag di dati per gestire le proprietà (o i parametri) degli oggetti in modo più semplice rispetto all'utilizzo di PCF (Programmable Command Format).

MQAI esegue le attività di amministrazione su un gestore code mediante l'utilizzo di *bag di dati*. I bag di dati consentono di gestire le proprietà (o i parametri) degli oggetti in modo più semplice rispetto all'utilizzo di PCF.

I vantaggi dell'utilizzo di MQAI sono i seguenti:

### Semplificare l'utilizzo dei messaggi PCF

MQAI è un modo più semplice per gestire IBM MQ. Se si utilizza MQAI, non è necessario scrivere i propri messaggi PCF. Ciò evita i problemi associati alle strutture dati complesse.

Per passare i parametri nei programmi scritti utilizzando le chiamate MQI, il messaggio PCF deve contenere il comando e i dettagli della stringa o dei dati interi. Per creare questa configurazione manualmente, è necessario aggiungere diverse istruzioni nel programma per ogni struttura e allocare spazio di memoria. Questo compito può essere lungo e laborioso.

I programmi scritti utilizzando MQAI inoltrano i parametri nella serie di dati appropriata ed è necessaria solo un'istruzione per ciascuna struttura. L'utilizzo dei sacchetti di dati MQAI elimina la necessità di gestire schiere e allocare memoria e fornisce un certo grado di isolamento dai dettagli del PCF.

### Gestire più facilmente le condizioni di errore

È difficile ottenere codici di ritorno dai comandi PCF. MQAI rende più semplice per il programma gestire le condizioni di errore.

## Scambio di dati tra applicazioni

I dati dell'applicazione vengono inviati in formato PCF e compressi e decompressi da MQAI. Se i dati del messaggio sono costituiti da numeri interi e stringhe di caratteri, è possibile utilizzare MQAI per sfruttare la conversione dei dati integrati IBM MQ per i dati PCF. Ciò evita la necessità di scrivere uscite di conversione dati.

Una volta creato e popolato il contenitore dati, è possibile inviare un messaggio di comando di gestione al server dei comandi di un gestore code, utilizzando la chiamata `mqExecute`. Questa chiamata attende i messaggi di risposta. La chiamata `mqExecute` gestisce lo scambio con il server dei comandi e restituisce le risposte in una *serie di risposte*.

## Esempi di utilizzo di MQAI

I seguenti programmi di esempio dimostrano l'utilizzo di MQAI per eseguire le diverse attività:

- `amqsaicq.c`: creare una coda locale.
- `amqsaicm.c`: visualizza gli eventi sullo schermo utilizzando un semplice controllo eventi.
- `amqsailq.c`: stampare un elenco di tutte le code locali e le relative profondità correnti.
- `amqsaicl.c`: stampa un elenco di tutti i canali e dei relativi tipi.

## Creazione dell'applicazione MQAI

Per creare la propria applicazione utilizzando MQAI, è necessario collegarsi alle stesse librerie di IBM MQ. Per informazioni su come creare le applicazioni IBM MQ, consultare [Creazione di un'applicazione procedurale](#).

## Suggerimenti e consigli per configurare IBM MQ utilizzando MQAI

MQAI utilizza i messaggi PCF per inviare i comandi di gestione al server dei comandi anziché gestire direttamente il server dei comandi stesso. Suggerimenti per la configurazione di IBM MQ utilizzando MQAI sono disponibili in [“Suggerimenti e consigli per l'utilizzo di MQAI per configurare IBM MQ”](#) a pagina 37.

### Riferimenti correlati

[IBM MQ Administration Interface](#)

## Suggerimenti e consigli per l'utilizzo di MQAI per configurare IBM MQ

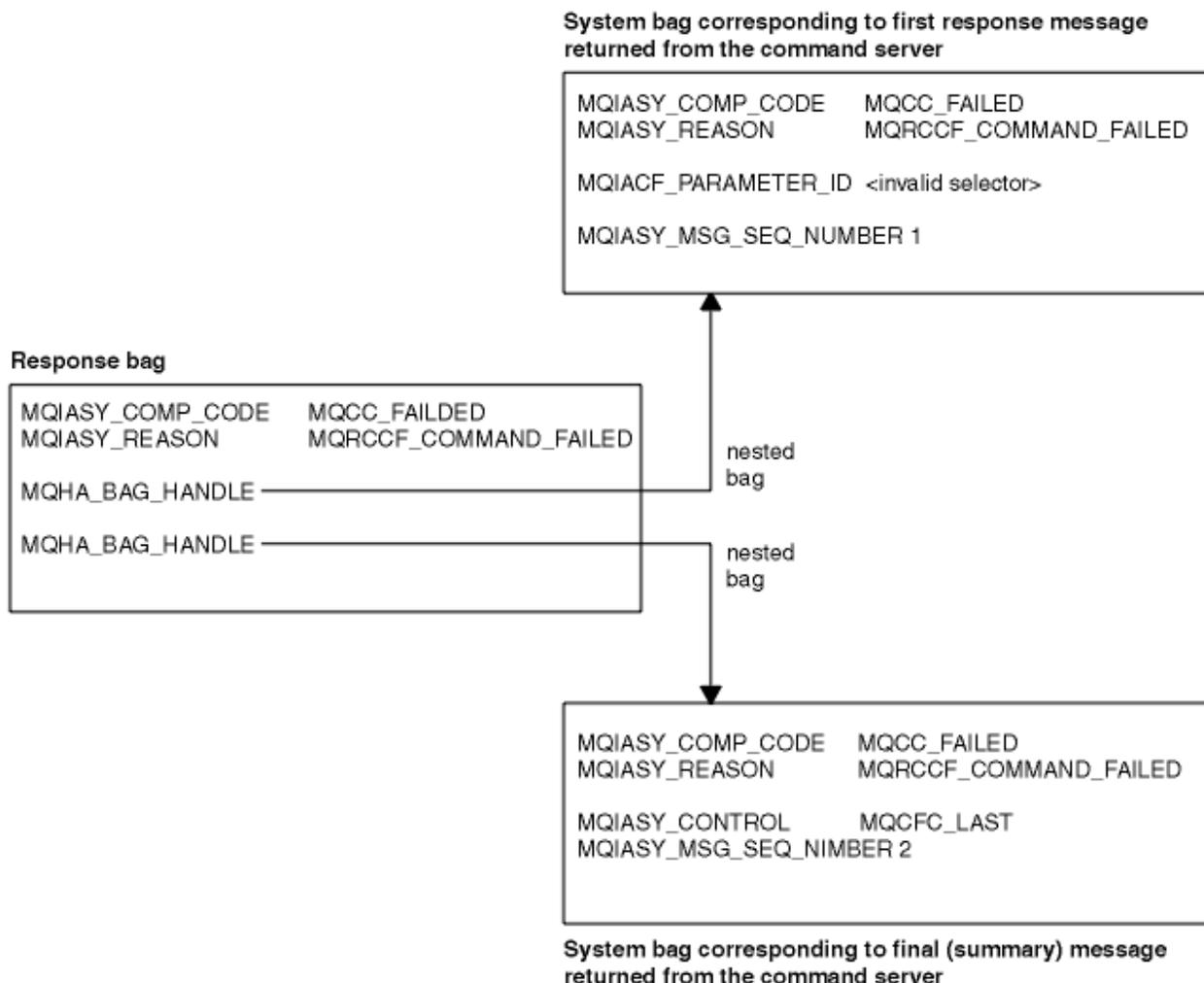
MQAI ( IBM MQ Administration Interface) utilizza i messaggi PCF per inviare i comandi di gestione al server dei comandi piuttosto che gestire direttamente il server dei comandi stesso. Di seguito sono riportati alcuni consigli per configurare IBM MQ utilizzando MQAI.

- Le stringhe di caratteri in IBM MQ sono spazi vuoti riempiti a lunghezza fissa. Utilizzando C, le stringhe con terminazione null possono normalmente essere fornite come parametri di immissione per le interfacce di programmazione IBM MQ .
- Per cancellare il valore di un attributo stringa, impostarlo su un singolo spazio piuttosto che su una stringa vuota.
- Considerare in anticipo gli attributi che si desidera modificare e analizzare solo questi attributi.
- Alcuni attributi non possono essere modificati, ad esempio un nome di coda o un tipo di canale. Accertarsi di tentare di modificare solo gli attributi che possono essere modificati. Fare riferimento all'elenco di parametri obbligatori e facoltativi per l'oggetto di modifica PCF specifico. Consultare [Definizioni dei formati dei comandi programmabili](#).
- Se una chiamata MQAI ha esito negativo, alcuni dettagli dell'errore vengono restituiti al pacchetto di risposte. Ulteriori dettagli possono essere trovati in un contenitore nidificato a cui può accedere il selettore `MQHA_BAG_HANDLE`. Ad esempio, se una chiamata `mqExecute` ha esito negativo con un codice di errore `MQRCCF_COMMAND_FAILED`, queste informazioni vengono restituite nella serie di risposte. Un motivo possibile per questo codice di errore è che un selettore specificato non era

valido per il tipo di messaggio di comando e questo dettaglio di informazioni si trova in un contenitore nidificato a cui può accedere un gestore contenitore.

Per ulteriori informazioni su MQExecute, consultare [“Invio dei comandi di gestione al server dei comandi qm utilizzando la chiamata mqExecute”](#) a pagina 72

Il seguente diagramma mostra questo scenario:



## Multi Argomenti MQAI avanzati

Informazioni sull'indicizzazione, sulla conversione dei dati e sull'uso del descrittore del messaggio

### Indicizzazione

Gli indici vengono utilizzati durante la sostituzione o la rimozione di elementi di dati esistenti da un contenitore per preservare l'ordine di inserimento.

### Conversione dati

Le stringhe contenute in un contenitore di dati MQAI possono essere in una varietà di serie di caratteri codificati e possono essere convertite utilizzando la chiamata [mqSetInteger](#).

### Utilizzo del descrittore del messaggio

MQAI genera un descrittore di messaggi impostato su un valore iniziale quando viene creato il contenitore di dati.

**Multi** **Indicizzazione in MQAI**

Gli indici vengono utilizzati durante la sostituzione o la rimozione di elementi dati esistenti da un contenitore. Esistono tre tipi di indicizzazione, che consentono di richiamare facilmente gli elementi di dati.

Ogni selettore e valore in un elemento dati in un contenitore ha tre numeri di indice associati:

- L'indice relativo ad altri elementi che hanno lo stesso selettore.
- L'indice relativo alla categoria del selettore (utente o sistema) a cui appartiene l'elemento.
- L'indice relativo a tutti gli elementi dati nel contenitore (utente e sistema).

Ciò consente l'indicizzazione da parte dei selettori utente, dei selettori di sistema o di entrambi, come mostrato in [Figura 3 a pagina 39](#).

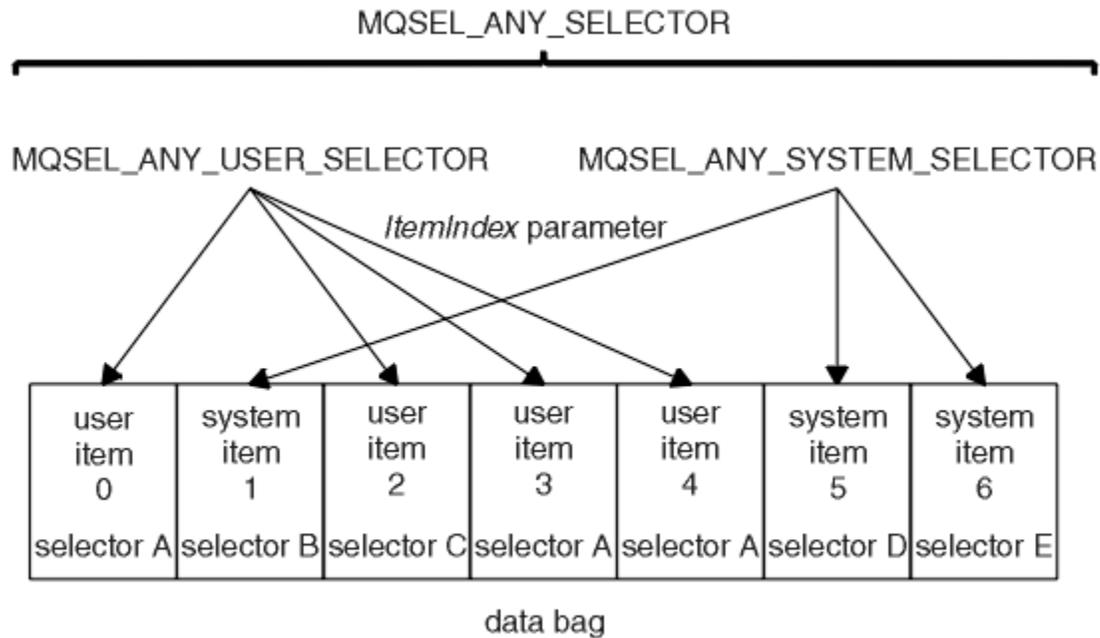


Figura 3. Indicizzazione

In [Figura 3 a pagina 39](#), l'elemento utente 3 (selettore A) può essere indicato dalle seguenti coppie di indici:

- selettore A (ItemIndex 1)
- MQSEL\_ANY\_USER\_SELECTOR (indice 2)
- MQSEL\_ANY\_SELECTOR (itemIndex 3)

L'indice è basato su zero come un array in C; se ci sono 'n' ricorrenze, l'indice varia da zero a 'n-1', senza interruzioni.

Gli indici vengono utilizzati durante la sostituzione o la rimozione di elementi dati esistenti da un contenitore. Quando viene utilizzato in questo modo, l'ordine di inserimento viene conservato, ma gli indici di altri elementi dati possono essere influenzati. Per esempi, consultare [“Modifica delle informazioni all'interno di un contenitore” a pagina 68](#) e [“Eliminazione di elementi dati” a pagina 71](#).

I tre tipi di indicizzazione consentono un facile richiamo degli elementi dati. Ad esempio, se ci sono tre istanze di un particolare selettore in un contenitore, la chiamata mqCountItems può contare il numero di istanze di tale selettore e le chiamate mqInquire \* possono specificare sia il selettore che l'indice per interrogare solo tali valori. Ciò è utile per gli attributi che possono avere un elenco di valori come alcune delle uscite sui canali.

Le stringhe contenute in un contenitore di dati MQAI possono essere in una varietà di serie di caratteri codificati. Queste stringhe possono essere convertite utilizzando la chiamata numero intero mqSet.

Come i messaggi PCF, le stringhe contenute in un contenitore di dati MQAI possono trovarsi in una varietà di serie di caratteri codificati. Di solito, tutte le stringhe in un messaggio PCF si trovano nella stessa serie di caratteri codificati, ovvero la stessa serie del gestore code.

Ogni elemento stringa in un contenitore di dati contiene due valori; la stringa stessa e il CCSID.

La stringa aggiunta al contenitore si ottiene dal parametro **Buffer** della chiamata `mqAddString` o `mqSetString`. Il CCSID viene ottenuto dall'elemento di sistema contenente un selettore di MQIASY\_CODED\_CHAR\_SET\_ID. Questo è noto come *bag CCSID* e può essere modificato utilizzando la chiamata `mqSetInteger`.

Quando si interroga il valore di una stringa contenuta in un contenitore di dati, il CCSID è un parametro di emissione dalla chiamata.

Tabella 3 a pagina 40 mostra le regole applicate durante la conversione dei bag di dati in messaggi e viceversa:

<i>Tabella 3. Elaborazione CCSID</i>			
<b>Chiamata MQAI</b>	<b>CCSID</b>	<b>Input da richiamare</b>	<b>Output da richiamare</b>
<code>mqBagToBuffer</code>	CCSID serie ( <u>1</u> )	Ignorato	Non modificato
<code>mqBagToBuffer</code>	CCSID stringa nel contenitore	Utilizzato	Non modificato
<code>mqBagToBuffer</code>	CCSID stringa nel buffer	Non applicabile	Copiato da CCSID stringa nel contenitore
<code>mqBufferToBag</code>	CCSID serie ( <u>1</u> )	Ignorato	Non modificato
<code>mqBufferToBag</code>	CCSID stringa nel buffer	Utilizzato	Non modificato
<code>mqBufferToBagmqBufferToBag</code>	CCSID stringa nel contenitore	Non applicabile	Copiato da CCSID stringa nel buffer
<code>mqPutSacchetto</code>	CCSID MQMD	Utilizzato	Non modificato ( <u>2</u> )
<code>mqPutSacchetto</code>	CCSID serie ( <u>1</u> )	Ignorato	Non modificato
<code>mqPutSacchetto</code>	CCSID stringa nel contenitore	Utilizzato	Non modificato
<code>mqPutSacchetto</code>	CCSID stringa nel messaggio inviato	Non applicabile	Copiato da CCSID stringa nel contenitore
<code>mqGetBag</code>	CCSID MQMD	Utilizzato per la conversione dei dati del messaggio	Impostato su CCSID dei dati restituiti ( <u>3</u> )
<code>mqGetBag</code>	CCSID serie ( <u>1</u> )	Ignorato	Non modificato
<code>mqGetBag</code>	CCSID stringa nel messaggio	Utilizzato	Non modificato
<code>mqGetBag</code>	CCSID stringa nel contenitore	Non applicabile	Copiato dai CCSID stringa nel messaggio
<code>mqExecute</code>	CCSID della serie di richiesta	Utilizzato per MQMD del messaggio di richiesta ( <u>4</u> )	Non modificato

Chiamata MQAI	CCSID	Input da richiamare	Output da richiamare
<a href="#">mqExecute</a>	CCSID della serie di risposte	Utilizzato per la conversione dati del messaggio di risposta (4)	Impostato su CCSID dei dati restituiti (3)
<a href="#">mqExecute</a>	CCSID stringa nella serie di richieste	Utilizzato per il messaggio di richiesta	Non modificato
<a href="#">mqExecute</a>	CCSID stringa nel bag di risposta	Non applicabile	Copiato dai CCSID stringa nel messaggio di risposta

**Note:**

1. Bag CCSID è l'elemento di sistema con selettore MQIASY\_CODED\_CHAR\_SET\_ID.
2. MQCCSI\_Q\_MGR è stato modificato nel CCSID del gestore code effettivo.
3. Se viene richiesta la conversione dei dati, il CCSID dei dati restituiti è uguale al valore di emissione. Se la conversione dati non è richiesta, il CCSID dei dati restituiti è uguale al valore del messaggio. Notare che non viene restituito alcun messaggio se la conversione dei dati è richiesta ma non riesce.
4. Se il CCSID è MQCCSI\_DEFAULT, viene utilizzato il CCSID del gestore code.

**Concetti correlati**

“Conversione dati tra serie di caratteri codificati” a pagina 195

I dati dei messaggi nei formati definiti da IBM MQ (noti anche come formati integrati) possono essere convertiti dal gestore code da una serie di caratteri codificati a un'altra, a condizione che entrambe le serie di caratteri siano relative a una singola lingua o a un gruppo di lingue simili.

“Il file ccsid\_part2.tbl” a pagina 197

Il file ccsid\_part2.tbl viene utilizzato per fornire ulteriori informazioni CCSID. Il file ccsid\_part2.tbl sostituisce il file ccsid.tbl utilizzato prima di IBM MQ 9.0.

**Multi Utilizzo del descrittore del messaggio in MQAI**

Il descrittore del messaggio generato da MQAI viene impostato su un valore iniziale quando viene creato il contenitore di dati.

Il tipo di comando PCF viene ottenuto dalla voce di sistema con il selettore MQIASY\_TYPE. Quando si crea un contenitore dati, il valore iniziale di questo elemento viene impostato in funzione del tipo di contenitore creato:

Tipo di sacchetto	Valore iniziale dell'elemento MQIASY_TYPE
BAG MQCBO_ADMIN_	COMANDO MQCFT
BAG MQCBO_COMMAND_	COMANDO MQCFT
MQCBO_*	UTENTE MQCFT

Quando MQAI genera un descrittore di messaggi, i valori utilizzati nei parametri **Format** e **MsgType** dipendono dal valore dell'elemento del sistema con il selettore MQIASY\_TYPE come mostrato in [Tabella 4](#) a pagina 41.

Tipo di comando PCF	Formato	MsgType
COMANDO MQCFT	MMQFMT_ADMIN	MQMT_REQUEST

Tipo di comando PCF	Formato	MsgType
REPORT MQCFT	MMQFMT_ADMIN	REPORT MQMT
MQCF_XX_ENCODE_CASE_ONE risposta	MMQFMT_ADMIN	MQMT_REPLY
MQCF_TRACE_ROUTE	MMQFMT_ADMIN	MQMT_DATAGRAM
EVENTO MQCFT	EVENTO MQFMT	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Tabella 5 a pagina 41 mostra che, se si crea un contenitore di gestione o un contenitore di comandi, il *Format* del descrittore del messaggio è MQFMT\_ADMIN e il *MsgType* è MQMT\_REQUEST. Questo è adatto per un messaggio di richiesta PCF inviato al server dei comandi quando è prevista una risposta.

Altri parametri nel descrittore del messaggio assumono i valori mostrati in Tabella 6 a pagina 42.

Parametro	Valore
<i>StrucId</i>	ID_STRUC_MQMD
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	Consultare Tabella 5 a pagina 41
<i>Expiry</i>	30 secondi (nota "1" a pagina 42)
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQEN_NATIVE
<i>CodedCharSetId</i>	dipende dal CCSID della serie (nota "2" a pagina 42)
<i>Format</i>	Consultare Tabella 5 a pagina 41
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	MQMI_NONE
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	vedere nota "3" a pagina 42
<i>ReplyToQMgr</i>	Spazio

**Note:**

1. Questo valore può essere sovrascritto nella chiamata mqExecute utilizzando il parametro **OptionsBag**. Per informazioni, consultare mqExecute.
2. Consultare "Elaborazione di conversione dati in MQAI" a pagina 40.
3. Nome della coda di risposta specificata dall'utente o della coda dinamica temporanea generata da MQAI per i messaggi di tipo MQMT\_REQUEST. In caso contrario, lasciare vuoto.

## Multi Programma C di esempio per la creazione di una coda locale (amqsaicq.c)

Il programma C di esempio amqsaicq.c crea una coda locale utilizzando MQAI.

```
/*
/*****
*/
/* Program name: AMQSAICQ.C */
/*
/* Description: Sample C program to create a local queue using the */
/* IBM MQ Administration Interface (MQAI). */
/*
/* Statement: Licensed Materials - Property of IBM */
/*
/* 84H2000, 5765-B73 */
/* 84H2001, 5639-B42 */
/* 84H2002, 5765-B74 */
/* 84H2003, 5765-B75 */
/* 84H2004, 5639-B43 */
/*
/* (C) Copyright IBM Corp. 1999, 2024. */
/*****
*/
/* Function: */
/* AMQSAICQ is a sample C program that creates a local queue and is an */
/* example of the use of the mqExecute call. */
/*
/* - The name of the queue to be created is a parameter to the program. */
/*
/* - A PCF command is built by placing items into an MQAI bag. */
/* These are:- */
/* - The name of the queue */
/* - The type of queue required, which, in this case, is local. */
/*
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q. */
/* The call generates the correct PCF structure. */
/* The call receives the reply from the command server and formats into */
/* the response bag. */
/*
/* - The completion code from the mqExecute call is checked and if there */
/* is a failure from the command server then the code returned by the */
/* command server is retrieved from the system bag that is */
/* embedded in the response bag to the mqExecute call. */
/*
/* Note: The command server must be running. */
/*
/*
/*****
*/
/* AMQSAICQ has 2 parameters - the name of the local queue to be created */
/* - the queue manager name (optional) */
/*****
*/
/* Includes */
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

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

void CheckCallResult(MQCHAR *, MQLONG , MQLONG );
void CreateLocalQueue(MQHCONN, MQCHAR *);

int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to IBM MQ connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQLONG reason; /* reason code */
}
```

```

/*****
/* First check the required parameters */
/*****
printf("Sample Program to Create a Local Queue\n");
if (argc < 2)
{
    printf("Required parameter missing - local queue name\n");
    exit(99);
}

/*****
/* Connect to the queue manager */
/*****
if (argc > 2)
    strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(QMName, &hConn, &compCode, &connReason);

/*****
/* Report reason and stop if connection failed */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("MQCONN", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Call the routine to create a local queue, passing the handle to the
/* queue manager and also passing the name of the queue to be created.
/*
/*****
CreateLocalQueue(hConn, argv[1]);

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
}
return 0;

}

/*****
/*
/* Function:      CreateLocalQueue
/* Description:  Create a local queue by sending a PCF command to the command
/*               server.
/*
/*****
/*
/* Input Parameters:  Handle to the queue manager
/*                   Name of the queue to be created
/*
/* Output Parameters: None
/*
/* Logic: The mqExecute call is executed with the command MQCMD_CREATE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply is read from the temporary queue and formatted into the
/* response bag.
/*
/* The completion code from the mqExecute call is checked and if there
/* is a failure from the command server then the code returned by the
/* command server is retrieved from the system bag that is
/* embedded in the response bag to the mqExecute call.
/*
/*****
void CreateLocalQueue(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG reason;                /* reason code */
    MQLONG compCode;              /* completion code */
    MQHBAG commandBag = MQHB_UNUSABLE_HBAG; /* command bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG resultBag;            /* result bag from mqExecute */
    MQLONG mqExecuteCC;          /* mqExecute completion code */
    MQLONG mqExecuteRC;          /* mqExecute reason code */
}

```

```

printf("\nCreating Local Queue %s\n\n", qName);

/*****
/* Create a command Bag for the mqExecute call. Exit the function if the
/* create fails.
*****/
mqCreateBag(MQCBO_ADMIN_BAG, &commandBag, &compCode, &reason);
CheckCallResult("Create the command bag", compCode, reason);
if (compCode !=MQCC_OK)
    return;

/*****
/* Create a response Bag for the mqExecute call, exit the function if the
/* create fails.
*****/
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create the response bag", compCode, reason);
if (compCode !=MQCC_OK)
    return;

/*****
/* Put the name of the queue to be created into the command bag. This will
/* be used by the mqExecute call.
*****/
mqAddString(commandBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, qName, &compCode,
            &reason);
CheckCallResult("Add q name to command bag", compCode, reason);

/*****
/* Put queue type of local into the command bag. This will be used by the
/* mqExecute call.
*****/
mqAddInteger(commandBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type to command bag", compCode, reason);

/*****
/* Send the command to create the required local queue.
/* The mqExecute call will create the PCF structure required, send it to
/* the command server and receive the reply from the command server into
/* the response bag.
*****/
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_CREATE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          commandBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response*/
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE*/
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

if (reason == MQRCD_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n")
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call and find the error if it failed.
*****/
if ( compCode == MQCC_OK )
    printf("Local queue %s successfully created\n", qName);
else
{
    printf("Creation of local queue %s failed: Completion Code = %d
           qName, compCode, reason);
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        /*****
        /* Get the system bag handle out of the mqExecute response bag.
        /* This bag contains the reason from the command server why the
        /* command failed.
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &resultBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****

```

```

        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /*****
mqInquireInteger(resultBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
                compCode, reason);
mqInquireInteger(resultBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag", compCode,
                reason);
        printf("Error returned by the command server: Completion code = %d :
                Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}
/*****
/* Delete the command bag if successfully created. */
/*****
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
    CheckCallResult("Delete the command bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}
} /* end of CreateLocalQueue */

/*****
/*
/* Function: CheckCallResult */
/*
/*
/*****
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
/*
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d :
                Reason = %d\n", callText, cc, rc);
}
}

```

## **Multi** Programma C di esempio per la visualizzazione di eventi utilizzando un controllo eventi (amqsaiem.c)

Il programma C di esempio amqsaiem.c illustra un controllo eventi di base utilizzando MQAI.

```

/*****
/*
/* Program name: AMQSAIEM.C */
/*
/* Description: Sample C program to demonstrate a basic event monitor */
/* using the IBM MQ Admin Interface (MQAI). */
/* Licensed Materials - Property of IBM */
/*
/*
/* 63H9336 */
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved. */
/*
/*
/* US Government Users Restricted Rights - Use, duplication or */
/* disclosure restricted by GSA ADP Schedule Contract with */

```

```

/* IBM Corp. */
/*****
/*
/* Function: */
/* AMQSAIEM is a sample C program that demonstrates how to write a simple */
/* event monitor using the mqGetBag call and other MQAI calls. */
/*
/* The name of the event queue to be monitored is passed as a parameter */
/* to the program. This would usually be one of the system event queues:- */
/* SYSTEM.ADMIN.QMGR.EVENT Queue Manager events */
/* SYSTEM.ADMIN.PERFM.EVENT Performance events */
/* SYSTEM.ADMIN.CHANNEL.EVENT Channel events */
/* SYSTEM.ADMIN.LOGGER.EVENT Logger events */
/*
/* To monitor the queue manager event queue or the performance event queue, */
/* the attributes of the queue manager need to be changed to enable */
/* these events. For more information about this, see Part 1 of the */
/* Programmable System Management book. The queue manager attributes can */
/* be changed using either MQSC commands or the MQAI interface. */
/* Channel events are enabled by default. */
/*
/* Program logic */
/* Connect to the Queue Manager. */
/* Open the requested event queue with a wait interval of 30 seconds. */
/* Wait for a message, and when it arrives get the message from the queue */
/* and format it into an MQAI bag using the mqGetBag call. */
/* There are many types of event messages and it is beyond the scope of */
/* this sample to program for all event messages. Instead the program */
/* prints out the contents of the formatted bag. */
/* Loop around to wait for another message until either there is an error */
/* or the wait interval of 30 seconds is reached. */
/*
/*****
/*
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored */
/* - the queue manager name (optional) */
/*
/*****

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

#include <cmqc.h> /* MQI */
#include <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */

/*****
/* Macros */
/*****
#if MQAT_DEFAULT == MQAT_WINDOWS_NT
#define Int64 "I64"
#elif defined(MQ_64_BIT)
#define Int64 "l"
#else
#define Int64 "ll"
#endif

/*****
/* Function prototypes */
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
void GetQEvents(MQHCONN, MQCHAR *);
int PrintBag(MQHBAG);
int PrintBagContents(MQHBAG, int);

/*****
/* Function: main */
/*****
int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QM name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */

/*****

```

```

/* First check the required parameters */
/*****
printf("Sample Event Monitor (times out after 30 secs)\n");
if (argc < 2)
{
    printf("Required parameter missing - event queue to be monitored\n");
    exit(99);
}

/*****
/* Connect to the queue manager */
/*****
if (argc > 2)
    strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(QMName, &hConn, &compCode, &connReason);
/*****
/* Report the reason and stop if the connection failed */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("MQCONN", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Call the routine to open the event queue and format any event messages */
/* read from the queue. */
/*****
GetQEvents(hConn, argv[1]);

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("MQDISC", compCode, reason);
}

return 0;
}

/*****
/*
/* Function: CheckCallResult */
/*
/*****
/*
/* Input Parameters:  Description of call */
/*                   Completion code */
/*                   Reason code */
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/*        reason code if the completion code is not successful */
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

/*****
/*
/* Function: GetQEvents */
/*
/*****
/*
/* Input Parameters:  Handle to the queue manager */
/*                   Name of the event queue to be monitored */
/*
/* Output Parameters: None */
/*
/* Logic:  Open the event queue. */
/*        Get a message off the event queue and format the message into */
/*        a bag. */
/*        A real event monitor would need to be programmed to deal with */

```

```

/*      each type of event that it receives from the queue. This is      */
/*      outside the scope of this sample, so instead, the contents of    */
/*      the bag are printed.                                             */
/*      The program waits for 30 seconds for an event message and then   */
/*      terminates if no more messages are available.                   */
/*      */
/*****
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason;          /* MQOPEN reason code      */
    MQLONG reason;             /* reason code             */
    MQLONG compCode;          /* completion code        */
    MQHOBJ eventQueue;        /* handle to event queue   */

    MQHBAG eventBag = MQHB_UNUSABLE_HBAG; /* event bag to receive event msg */
    MQOD od = {MQOD_DEFAULT}; /* Object Descriptor      */
    MQMD md = {MQMD_DEFAULT}; /* Message Descriptor     */
    MQGMO gmo = {MQGMO_DEFAULT}; /* get message options   */
    MQLONG bQueueOK = 1; /* keep reading msgs while true */

    /*****
    /* Create an Event Bag in which to receive the event.                */
    /* Exit the function if the create fails.                             */
    /*****
    mqCreateBag(MQCBO_USER_BAG, &eventBag, &compCode, &reason);
    CheckCallResult("Create event bag", compCode, reason);
    if (compCode != MQCC_OK)
        return;

    /*****
    /* Open the event queue chosen by the user                            */
    /*****
    strncpy(od.ObjectName, qName, (size_t)MQ_Q_NAME_LENGTH);
    MQOPEN(hConn, &od, MQOO_INPUT_AS_Q_DEF+MQOO_FAIL_IF QUIESCING, &eventQueue,
            &compCode, &openReason);
    CheckCallResult("Open event queue", compCode, openReason);

    /*****
    /* Set the GMO options to control the action of the get message from the */
    /* queue.                                                                    */
    /*****
    gmo.WaitInterval = 30000; /* 30 second wait for message */
    gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF QUIESCING + MQGMO_CONVERT;
    gmo.Version = MQGMO_VERSION_2; /* Avoid need to reset Message ID */
    gmo.MatchOptions = MQMO_NONE; /* and Correlation ID after every */
    /* mqGetBag

    /*****
    /* If open fails, we cannot access the queue and must stop the monitor. */
    /*****
    if (compCode != MQCC_OK)
        bQueueOK = 0;

    /*****
    /* Main loop to get an event message when it arrives                    */
    /*****
    while (bQueueOK)
    {
        printf("\nWaiting for an event\n");

        /*****
        /* Get the message from the event queue and convert it into the event */
        /* bag.                                                                    */
        /*****
        mqGetBag(hConn, eventQueue, &md, &gmo, eventBag, &compCode, &reason);

        /*****
        /* If get fails, we cannot access the queue and must stop the monitor. */
        /*****
        if (compCode != MQCC_OK)
        {
            bQueueOK = 0;

            /*****
            /* If get fails because no message available then we have timed out, */
            /* so report this, otherwise report an error.                            */
            /*****
            if (reason == MQRC_NO_MSG_AVAILABLE)
            {
                printf("No more messages\n");
            }
            else
            {

```

```

        CheckCallResult("Get bag", compCode, reason);
    }
}

/*****
/* Event message read - Print the contents of the event bag */
*****/
else
{
    if ( PrintBag(eventBag) )
        printf("\nError found while printing bag contents\n");

    } /* end of msg found */
} /* end of main loop */
/*****
/* Close the event queue if successfully opened */
*****/
if (openReason == MQRC_NONE)
{
    MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
    CheckCallResult("Close event queue", compCode, reason);
}

/*****
/* Delete the event bag if successfully created. */
*****/
if (eventBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&eventBag, &compCode, &reason);
    CheckCallResult("Delete the event bag", compCode, reason);
}
} /* end of GetQEvents */

/*****
/*
/* Function: PrintBag
/*
*****/
/*
/* Input Parameters: Bag Handle
/*
/*
/* Output Parameters: None
/*
/*
/* Returns: Number of errors found
/*
/*
/* Logic: Calls PrintBagContents to display the contents of the bag.
/*
*****/

int PrintBag(MQHBAG dataBag)
{
    int errors;

    printf("\n");
    errors = PrintBagContents(dataBag, 0);
    printf("\n");

    return errors;
}

/*****
/*
/* Function: PrintBagContents
/*
*****/
/*
/* Input Parameters: Bag Handle
/*
/* Indentation level of bag
/*
/*
/* Output Parameters: None
/*
/*
/* Returns: Number of errors found
/*
/*
/* Logic: Count the number of items in the bag
/*
/* Obtain selector and item type for each item in the bag.
/*
/* Obtain the value of the item depending on item type and display the
/*
/* index of the item, the selector and the value.
/*
/* If the item is an embedded bag handle then call this function again
/*
/* to print the contents of the embedded bag increasing the
/*
/* indentation level.
*****/

```

```

/*****
int PrintBagContents(MQHBAG dataBag, int indent)
{
    /*****/
    /* Definitions */
    /*****/
    #define LENGTH 500 /* Max length of string to be read*/
    #define INDENT 4 /* Number of spaces to indent */
    /* embedded bag display */

    /*****/
    /* Variables */
    /*****/
    MQLONG itemCount; /* Number of items in the bag */
    MQLONG itemType; /* Type of the item */
    int i; /* Index of item in the bag */
    MQCHAR stringVal[LENGTH+1]; /* Value if item is a string */
    MQBYTE byteStringVal[LENGTH]; /* Value if item is a byte string */
    MQLONG stringLength; /* Length of string value */
    MQLONG ccsid; /* CCSID of string value */
    MQINT32 iValue; /* Value if item is an integer */
    MQINT64 i64Value; /* Value if item is a 64-bit
    /* integer */

    MQLONG selector; /* Selector of item */
    MQHBAG bagHandle; /* Value if item is a bag handle */
    MQLONG reason; /* reason code */
    MQLONG compCode; /* completion code */
    MQLONG trimLength; /* Length of string to be trimmed */
    int errors = 0; /* Count of errors found */
    char blanks[] = " /* Blank string used to
    /* indent display */

    /*****/
    /* Count the number of items in the bag */
    /*****/
    mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

    if (compCode != MQCC_OK)
        errors++;
    else
    {
        printf("
        printf("
        printf("
    }

    /*****/
    /* If no errors found, display each item in the bag */
    /*****/
    if (!errors)
    {
        for (i = 0; i < itemCount; i++)
        {
            /*****/
            /* First inquire the type of the item for each item in the bag */
            /*****/
            mqInquireItemInfo(dataBag, /* Bag handle */
                MQSEL_ANY_SELECTOR, /* Item can have any selector*/
                i, /* Index position in the bag */
                &selector, /* Actual value of selector */
                /* returned by call */
                &itemType, /* Actual type of item */
                /* returned by call */
                &compCode, /* Completion code */
                &reason); /* Reason Code */

            if (compCode != MQCC_OK)
                errors++;

            switch(itemType)
            {
            case MQITEM_INTEGER:
                /*****/
                /* Item is an integer. Find its value and display its index, */
                /* selector and value. */
                /*****/
                mqInquireInteger(dataBag, /* Bag handle */
                    MQSEL_ANY_SELECTOR, /* Allow any selector */
                    i, /* Index position in the bag */
                    &iValue, /* Returned integer value

```

```

                                &compCode,      /* Completion code      */
                                &reason);        /* Reason Code          */

if (compCode != MQCC_OK)
    errors++;
else
    printf("%.s %-2d %-4d (%d)\n",
           indent, blanks, i, selector, iValue);
break

case MQITEM_INTEGER64:
/*****
/* Item is a 64-bit integer. Find its value and display its
/* index, selector and value.
*****/
mqInquireInteger64(dataBag, /* Bag handle
MQSEL_ANY_SELECTOR, /* Allow any selector
i, /* Index position in the bag
&i64Value, /* Returned integer value
&compCode, /* Completion code
&reason); /* Reason Code

if (compCode != MQCC_OK)
    errors++;
else
    printf("%.s %-2d %-4d (%"Int64"d)\n",
           indent, blanks, i, selector, i64Value);
break;

case MQITEM_STRING:
/*****
/* Item is a string. Obtain the string in a buffer, prepare
/* the string for displaying and display the index, selector,
/* string and Character Set ID.
*****/
mqInquireString(dataBag, /* Bag handle
MQSEL_ANY_SELECTOR, /* Allow any selector
i, /* Index position in the bag
LENGTH, /* Maximum length of buffer
stringVal, /* Buffer to receive string
&stringLength, /* Actual length of string
&ccsid, /* Coded character set ID
&compCode, /* Completion code
&reason); /* Reason Code

/*****
/* The call can return a warning if the string is too long for
/* the output buffer and has been truncated, so only check
/* explicitly for call failure.
*****/
if (compCode == MQCC_FAILED)
    errors++;
else
{
/*****
/* Remove trailing blanks from the string and terminate with
/* a null. First check that the string should not have been
/* longer than the maximum buffer size allowed.
*****/
if (stringLength > LENGTH)
    trimLength = LENGTH;
else
    trimLength = stringLength;
mqTrim(trimLength, stringVal, stringVal, &compCode, &reason);
printf("%.s %-2d %-4d '%s' %d\n",
       indent, blanks, i, selector, stringVal, ccsid);
}
break;

case MQITEM_BYTE_STRING:
/*****
/* Item is a byte string. Obtain the byte string in a buffer,
/* prepare the byte string for displaying and display the
/* index, selector and string.
*****/
mqInquireByteString(dataBag, /* Bag handle
MQSEL_ANY_SELECTOR, /* Allow any selector
i, /* Index position in the bag
LENGTH, /* Maximum length of buffer
byteStringVal, /* Buffer to receive string
&stringLength, /* Actual length of string

```

```

                                &compCode, /* Completion code */
                                &reason); /* Reason Code */

/*****
/* The call can return a warning if the string is too long for */
/* the output buffer and has been truncated, so only check */
/* explicitly for call failure. */
/*****
if (compCode == MQCC_FAILED)
    errors++;
else
{
    printf("%.s %-2d %-4d X'",
           indent, blanks, i, selector);

    for (i = 0 ; i < stringLength ; i++)
        printf("

    printf("\n");
}
break;

case MQITEM_BAG:
/*****
/* Item is an embedded bag handle, so call the PrintBagContents*/
/* function again to display the contents. */
/*****
mqInquireBag(dataBag, /* Bag handle */
             MQSEL_ANY_SELECTOR, /* Allow any selector */
             i, /* Index position in the bag */
             &bagHandle, /* Returned embedded bag hdl*/
             &compCode, /* Completion code */
             &reason); /* Reason Code */

if (compCode != MQCC_OK)
    errors++;
else
{
    printf("%.s %-2d %-4d (%d)\n", indent, blanks, i,
           selector, bagHandle);
    if (selector == MQHA_BAG_HANDLE)
        printf("
    else
        printf("
        PrintBagContents(bagHandle, indent+INDENT);
}
break;

default:
    printf("
}
}
}
return errors;
}

```

## **Multi** Programma C di esempio per l'interrogazione degli oggetti canale (amqsaicl.c)

Il programma C di esempio amqsaicl.c interroga gli oggetti canale utilizzando MQAI.

```

/*****
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/*                using the IBM MQ Administration Interface (MQAI)
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.

```

```

/* <NOC_COPYRIGHT> */
/*****
/*
/* Function: */
/* AMQSAICL is a sample C program that demonstrates how to inquire */
/* attributes of the local queue manager using the MQAI interface. In */
/* particular, it inquires all channels and their types. */
/*
/* - A PCF command is built from items placed into an MQAI administration */
/* bag. */
/* These are:- */
/* - The generic channel name "*" */
/* - The attributes to be inquired. In this sample we just want */
/* name and type attributes */
/*
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed. */
/* The call generates the correct PCF structure. */
/* The default options to the call are used so that the command is sent */
/* to the SYSTEM.ADMIN.COMMAND.QUEUE. */
/* The reply from the command server is placed on a temporary dynamic */
/* queue. */
/* The reply from the MQCMD_INQUIRE_CHANNEL is read from the */
/* temporary queue and formatted into the response bag. */
/*
/* - The completion code from the mqExecute call is checked and if there */
/* is a failure from the command server, then the code returned by the */
/* command server is retrieved from the system bag that has been */
/* embedded in the response bag to the mqExecute call. */
/*
/* Note: The command server must be running. */
/*
/*****
/*
/* AMQSAICL has 2 parameter - the queue manager name (optional) */
/* - output file (optional) default varies */
/*****

/*****
/* Includes */
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

#include <cmqc.h> /* MQI */
#include <cmqcf.h> /* PCF */
#include <cmqbc.h> /* MQAI */
#include <cmqxc.h> /* MQCD

/*****
/* Function prototypes */
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* DataTypes */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFIL FILE OUTFILEHDL;
#else
typedef FILE OUTFILEHDL;
#endif

/*****
/* Constants */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
char name[9];
} ChlTypeMap[9] =
{
" *SDR ", /* MQCHT_SENDER */
" *SVR ", /* MQCHT_SERVER */
" *RCVR ", /* MQCHT_RECEIVER */
" *RQSTR ", /* MQCHT_REQUESTER */
" *ALL ", /* MQCHT_ALL */
" *CLTCN ", /* MQCHT_CLNTCONN */

```

```

    "SVRCONN ", /* MQCHT_SVRCONN */
    "CLUSRCVR", /* MQCHT_CLUSRCVR */
    "CLUSSDR " /* MQCHT_CLUSSDR */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER */
    "svr      ", /* MQCHT_SERVER */
    "rcvr     ", /* MQCHT_RECEIVER */
    "rqstr    ", /* MQCHT_REQUESTER */
    "all      ", /* MQCHT_ALL */
    "cltconn  ", /* MQCHT_CLNTCONN */
    "svrcn    ", /* MQCHT_SVRCONN */
    "clusrcvr ", /* MQCHT_CLUSRCVR */
    "cluscdr  ", /* MQCHT_CLUSSDR */
};
#endif

/*****
/* Macros */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = _Ropen((fname), "wr", rtncode=Y");
#define CLOSEOUTFILE(hdl) \
    _Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _Rwrite((hdl), (buf), (buflen));

#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));

#else
#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(fname) \
    fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));

#endif

#define ChlType2String(t) ChlTypeMap[(t)-1].name

/*****
/* Function: main */
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables */
    /*****
    MQHCONN hConn; /* handle to MQ connection */
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG cAttrsBag; /* bag containing chl attributes */
    MQHBAG errorBag; /* bag containing cmd server error */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */
    MQLONG chlNameLength; /* Actual length of chl name */
    MQLONG chlType; /* Channel type */
    MQLONG i; /* loop counter */
    MQLONG numberOfBags; /* number of bags in response bag */
    MQCHAR chlName[MQ_OBJECT_NAME_LENGTH+1]; /* name of chl extracted from bag */
    MQCHAR OutputBuffer[100]; /* output data buffer */
    OUTFILEHDL *outfp = NULL; /* output file handle */

```

```

/*****
/* Connect to the queue manager */
/*****
if (argc > 1)
    strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(qmName, &hConn, &compCode, &connReason);

/*****
/* Report the reason and stop if the connection failed. */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("Queue Manager connection", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Open the output file */
/*****
if (argc > 2)
{
    OPENOUTFILE(outfp, argv[2]);
}
else
{
    OPENOUTFILE(outfp, OUTFILE);
}

if(outfp == NULL)
{
    printf("Could not open output file.\n");
    goto MOD_EXIT;
}

/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
CheckCallResult("Create admin bag", compCode, reason);

/*****
/* Create a response bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic channel name into the admin bag */
/*****
mqAddString(adminBag, MQCACH_CHANNEL_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode, &reason);
CheckCallResult("Add channel name", compCode, reason);

/*****
/* Put the channel type into the admin bag */
/*****
mqAddInteger(adminBag, MQIACH_CHANNEL_TYPE, MQCHT_ALL, &compCode, &reason);
CheckCallResult("Add channel type", compCode, reason);

/*****
/* Add an inquiry for various attributes */
/*****
mqAddInquiry(adminBag, MQIACH_CHANNEL_TYPE, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the channel names and channel types. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* MQ connection handle */
          MQCMD_INQUIRE_CHANNEL, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqexecute */
          &reason); /* Reason code from mqexecute call */

```

```

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName="">\n");
    goto MOD_EXIT;
}

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each channel are in separate bags. */
    /*****
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags;,
                &compCode;, &reason;);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfbags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the channel attributes */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &cAttrsBag,
                    &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the channel name out of the channel attributes bag */
        /*****
        mqInquireString(cAttrsBag, MQCACH_CHANNEL_NAME, 0, MQ_OBJECT_NAME_LENGTH,
                       chlName, &chlNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get channel name", compCode, reason);

        /*****
        /* Get the channel type out of the channel attributes bag */
        /*****
        mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                        &compCode, &reason);
        CheckCallResult("Get type", compCode, reason);

        /*****
        /* Use mqTrim to prepare the channel name for printing. */
        /* Print the result. */
        /*****
        mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
        sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
        WRITEOUTFILE(outfp,OutputBuffer,29)
    }
}

else /* Failed mqExecute */
{
    printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
           compCode, reason);
    /*****
    /* If the command fails get the system bag handle out of the mqexecute */
    /* response bag.This bag contains the reason from the command server */
    /* why the command failed. */
    /*****
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
                    &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /*****
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                        &compCode, &reason );
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,

```

```

        &compCode, &reason);
    CheckCallResult("Get the reason code from the result bag",
        compCode, reason);
    printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
        mqExecuteCC, mqExecuteRC);
}
}

MOD_EXIT:
/*****
/* Delete the admin bag if successfully created. */
/*****
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
}

/*****
/* Close the output file if open */
/*****
if(outfp != NULL)
    CLOSEOUTFILE(outfp);

return 0;
}

/*****
/*
/* Function: CheckCallResult */
/*
/*
/*****
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
/*
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %ld : Reason = %ld\n", callText,
            cc, rc);
}

```

## Multi Programma C di esempio per l'interrogazione delle code e delle informazioni di stampa (amqsailq.c)

Il programma C di esempio amqsailq.c interroga la profondità corrente delle code locali utilizzando MQAI.

```

/*****
/*

```

```

/* Program name: AMQSAILQ.C */
/*
/* Description: Sample C program to inquire the current depth of the local
/* queues using the IBM MQ Administration Interface (MQAI) */
/*
/* Statement: Licensed Materials - Property of IBM */
/*
/* 84H2000, 5765-B73 */
/* 84H2001, 5639-B42 */
/* 84H2002, 5765-B74 */
/* 84H2003, 5765-B75 */
/* 84H2004, 5639-B43 */
/*
/* (C) Copyright IBM Corp. 1999, 2024. */
/*
/*****
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/* bag.
/* These are:-
/* - The generic queue name "*"
/* - The type of queue required. In this sample we want to
/* inquire local queues.
/* - The attribute to be inquired. In this sample we want the
/* current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_Q command is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* - If the call is successful, the depth of each local queue is placed
/* in system bags embedded in the response bag of the mqExecute call.
/* The name and depth of each queue is obtained from each of the bags
/* and the result displayed on the screen.
/*
/* Note: The command server must be running.
/*
/*****
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
/*****

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

#include <cmqc.h> /* MQI */
#include <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */

/*****
/* Function prototypes
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
/*
/* MQAI variables
*/

```

```

/*****
MQHCONN hConn; /* handle to IBM MQ connection */
MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
MQLONG reason; /* reason code */
MQLONG connReason; /* MQCONN reason code */
MQLONG compCode; /* completion code */
MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
MQHBAG qAttrsBag; /* bag containing q attributes */
MQHBAG errorBag; /* bag containing cmd server error */
MQLONG mqExecuteCC; /* mqExecute completion code */
MQLONG mqExecuteRC; /* mqExecute reason code */
MQLONG qNameLength; /* Actual length of q name */
MQLONG qDepth; /* depth of queue */
MQLONG i; /* loop counter */
MQLONG numberOfBags; /* number of bags in response bag */
MQCHAR qName[MQ_Q_NAME_LENGTH+1]; /* name of queue extracted from bag*/

printf("Display current depths of local queues\n\n");

/*****
/* Connect to the queue manager */
/*****
if (argc > 1)
    strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
MQCONN(qmName, &hConn, &compCode, &connReason);

/*****
/* Report the reason and stop if the connection failed. */
/*****
if (compCode == MQCC_FAILED)
{
    CheckCallResult("Queue Manager connection", compCode, connReason);
    exit( (int)connReason);
}

/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
CheckCallResult("Create admin bag", compCode, reason);
/*****
/* Create a response bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
CheckCallResult("Create response bag", compCode, reason);

/*****
/* Put the generic queue name into the admin bag */
/*****
mqAddString(adminBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, "*",
            &compCode, &reason);
CheckCallResult("Add q name", compCode, reason);

/*****
/* Put the local queue type into the admin bag */
/*****
mqAddInteger(adminBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type", compCode, reason);

/*****
/* Add an inquiry for current queue depths */
/*****
mqAddInquiry(adminBag, MQIA_CURRENT_Q_DEPTH, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the local queue names and queue depths. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* IBM MQ connection handle */
          MQCMD_INQUIRE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */

```

```

        &reason);          /* Reason code from mqExecute call      */

/*****
/* Check the command server is started. If not exit.          */
*****/
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call. If successful find the current
/* depths of all the local queues. If failed find the error.
*****/
if ( compCode == MQCC_OK )          /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the
    /* mqExecute call. The attributes for each queue are in a separate bag.
    *****/
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
        &reason);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag.
        /* This bag contains the queue attributes
        *****/
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &qAttrsBag, &compCode,
            &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the queue name out of the queue attributes bag
        *****/
        mqInquireString(qAttrsBag, MQCA_Q_NAME, 0, MQ_Q_NAME_LENGTH, qName,
            &qNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get queue name", compCode, reason);

        /*****
        /* Get the depth out of the queue attributes bag
        *****/
        mqInquireInteger(qAttrsBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
            &compCode, &reason);
        CheckCallResult("Get depth", compCode, reason);

        /*****
        /* Use mqTrim to prepare the queue name for printing.
        /* Print the result.
        *****/
        mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason);
        printf("%4d %-48s\n", qDepth, qName);
    }
}

else          /* Failed mqExecute */
{
    printf("Call to get queue attributes failed: Completion Code = %d :
        Reason = %d\n", compCode, reason);

    /*****
    /* If the command fails get the system bag handle out of the mqExecute
    /* response bag. This bag contains the reason from the command server
    /* why the command failed.
    *****/
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag, &compCode,
            &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command
        /* server, from the embedded error bag.
        *****/
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,

```

```

        &compCode, &reason );
    CheckCallResult("Get the completion code from the result bag",
        compCode, reason);
    mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
        &compCode, &reason);
    CheckCallResult("Get the reason code from the result bag",
        compCode, reason);
    printf("Error returned by the command server: Completion Code = %d :
        Reason = %d\n", mqExecuteCC, mqExecuteRC);
}
}

/*****
/* Delete the admin bag if successfully created. */
/*****
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/*****
/* Disconnect from the queue manager if not already connected */
/*****
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from queue manager", compCode, reason);
}
return 0;
}

*****/
* Function: CheckCallResult */
* */
*****/
* Input Parameters: Description of call */
* Completion code */
* Reason code */
* */
* Output Parameters: None */
* */
* Logic: Display the description of the call, the completion code and the */
* reason code if the completion code is not successful */
* */
*****/
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}
}

```

## Multi Data bag e MQAI

Un contenitore di dati è un mezzo per gestire le proprietà o i parametri degli oggetti utilizzando MQAI (IBM MQ Administration Interface).

### Data bag

- La serie di dati contiene zero o più *elementi dati*. Questi elementi dati vengono ordinati all'interno del contenitore quando vengono inseriti nel contenitore. Questo è chiamato *ordine di inserimento*. Ogni elemento dati contiene un *selettore* che identifica l'elemento dati e un *valore* di tale elemento dati che

può essere un numero intero, un numero intero a 64 bit, un filtro di numeri interi, una stringa, un filtro di stringhe, una stringa di byte, un filtro di stringhe di byte o un handle di un altro contenitore. Gli elementi dati sono descritti in dettaglio in [“Tipi di elementi dati disponibili in MQAI”](#) a pagina 65

Esistono due tipi di selettori: *selettori utente* e *selettori di sistema*. Questi sono descritti in [Selettori MQAI](#). I selettori sono di solito univoci, ma è possibile avere più valori per lo stesso selettore. In questo caso, un *indice* identifica la particolare ricorrenza del selettore richiesta. Gli indici sono descritti in [“Indicizzazione in MQAI”](#) a pagina 39.

Una gerarchie di questi concetti è mostrata nella [Figura 1](#).

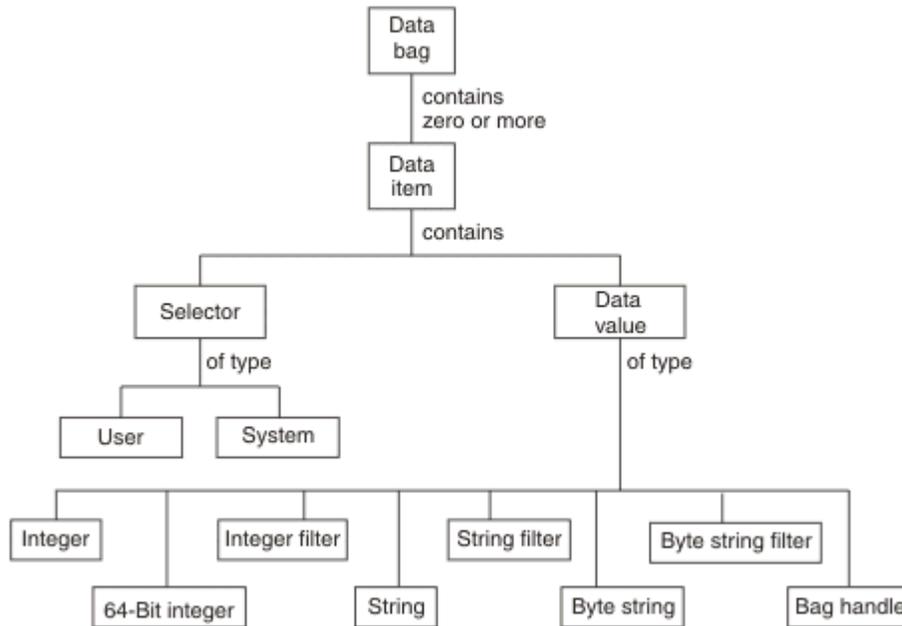


Figura 4. Gerarchia dei concetti MQAI

La gerarchia è stata illustrata in un paragrafo precedente.

## Tipi di serie di dati

È possibile scegliere il tipo di contenitore dati che si desidera creare in base all'attività che si desidera eseguire:

### serie utente

Un semplice contenitore utilizzato per i dati utente.

### serie di gestione

Un contenitore creato per i dati utilizzati per gestire gli oggetti IBM MQ inviando messaggi di gestione a un server dei comandi. Il contenitore di amministrazione implica automaticamente alcune opzioni come descritto in [“Creazione ed eliminazione di bag di dati”](#) a pagina 64.

### serie di comandi

Un contenitore creato anche per comandi per la gestione di oggetti IBM MQ . Tuttavia, a differenza della serie di amministrazione, la serie di comandi non implica automaticamente alcune opzioni, sebbene queste opzioni siano disponibili. Per ulteriori informazioni sulle opzioni, consultare [“Creazione ed eliminazione di bag di dati”](#) a pagina 64.

### gruppo

Un contenitore utilizzato per contenere una serie di elementi dati raggruppati. Le borse di gruppo non possono essere utilizzate per la gestione degli oggetti IBM MQ .

Inoltre, la **serie di sistemi** viene creata da MQAI quando un messaggio di risposta viene restituito dal server dei comandi e collocato in una serie di output dell'utente. Un contenitore di sistema non può essere modificato dall'utente.

I diversi modi di utilizzare i data bag sono elencati in questo argomento:

## Utilizzo dei data bag

I diversi modi di utilizzare i data bag sono mostrati nel seguente elenco:

- È possibile creare ed eliminare i data bag [“Creazione ed eliminazione di bag di dati”](#) a pagina 64.
- È possibile inviare dati tra le applicazioni utilizzando i data bag [“Inserimento e ricezione di bag di dati mediante MQAI”](#) a pagina 65.
- È possibile aggiungere elementi dati ai bag di dati [“Aggiunta di elementi dati ai bag con MQAI”](#) a pagina 66.
- È possibile aggiungere un comando di interrogazione all'interno di un contenitore dati [“Aggiunta di un comando di interrogazione ad un contenitore”](#) a pagina 67.
- È possibile richiedere informazioni all'interno dei data bag [“Richiesta all'interno dei bag di dati”](#) a pagina 68.
- È possibile contare gli elementi dati all'interno di una serie di dati [“Conteggio elementi dati”](#) a pagina 70.
- È possibile modificare le informazioni in un contenitore dati [“Modifica delle informazioni all'interno di un contenitore”](#) a pagina 68.
- È possibile cancellare un contenitore dati [“Cancellazione di un contenitore utilizzando la chiamata mqClearBag”](#) a pagina 69.
- È possibile troncare una serie di dati [“Troncamento di un sacchetto utilizzando la chiamata mqTruncateBag”](#) a pagina 69.
- È possibile convertire i bag e i buffer [“Conversione di buste e buffer”](#) a pagina 70.

Multi

### **Creazione ed eliminazione di bag di dati**

## Creazione di bag di dati

Per utilizzare MQAI, creare prima un contenitore di dati utilizzando la chiamata mqCreateBag. Come input per questa chiamata, fornire una o più opzioni per controllare la creazione della borsa.

Il parametro **Options** della chiamata MQCreateBag consente di scegliere se creare un contenitore utente, un contenitore comandi, un contenitore gruppi o un contenitore di amministrazione.

Per creare una serie di utenti, una serie di comandi o una serie di gruppi, è possibile scegliere una o più ulteriori opzioni per:

- Utilizzare il modulo di elenco quando ci sono due o più ricorrenze adiacenti dello stesso selettore in un contenitore.
- Riordinare le voci di dati man mano che vengono aggiunte a un messaggio PCF per garantire che i parametri siano nell'ordine corretto. Per ulteriori informazioni sugli elementi dati, consultare [“Tipi di elementi dati disponibili in MQAI”](#) a pagina 65.
- Controllare i valori dei selettori utente per gli elementi che si aggiungono al contenitore.

Le borse di amministrazione implicano automaticamente queste opzioni.

Una serie di dati è identificata dal relativo handle. La maniglia della borsa viene restituita da mqCreateBag e deve essere fornita su tutte le altre chiamate che utilizzano la borsa dati.

Per una descrizione completa della chiamata mqCreateBag, consultare [mqCreateBag](#).

## Eliminazione dei bag di dati

Qualsiasi contenitore di dati creato dall'utente deve essere eliminato anche utilizzando la chiamata `mqDeleteBag`. Ad esempio, se un contenitore viene creato nel codice utente, deve essere eliminato anche nel codice utente.

I bag di sistema vengono creati ed eliminati automaticamente da MQAI. Per ulteriori informazioni, consultare [“Invio dei comandi di gestione al server dei comandi qm utilizzando la chiamata `mqExecute`” a pagina 72](#). Il codice utente non può eliminare un contenitore di sistema.

Per una descrizione completa della chiamata `mqDeleteBag`, consultare [mqDeleteBag](#).

### **Multi** *Inserimento e ricezione di bag di dati mediante MQAI*

I dati possono anche essere inviati tra le applicazioni inserendo e ottenendo i bag di dati utilizzando le chiamate di borsa `mqPut` `mqGet`. Ciò consente a IBM MQ Administration Interface (MQAI) di gestire il buffer piuttosto che l'applicazione.

La chiamata al contenitore `mqPut` converte il contenuto del contenitore specificato in un messaggio PCF e invia il messaggio alla coda specificata e la chiamata al contenitore `mqGet` rimuove il messaggio dalla coda specificata e lo riconverte in un contenitore di dati. Pertanto, la chiamata al sacchetto `mqPut` è l'equivalente della chiamata `mqBagToBuffer` seguita da `MQPUT` e il sacchetto `mqGet` è l'equivalente della chiamata `MQGET` seguita da `mqBufferToBag`.

Per ulteriori informazioni sull'invio e la ricezione di messaggi PCF in una coda specifica, consultare [“Invio e ricezione di messaggi PCF in una coda specificata” a pagina 27](#)

**Nota:** Se si sceglie di utilizzare la chiamata del sacchetto `mqGet`, i dettagli PCF all'interno del messaggio devono essere corretti; in caso contrario, si verifica un errore appropriato e il messaggio PCF non viene restituito.

### **Multi** *Tipi di elementi dati disponibili in MQAI*

Gli elementi dati vengono utilizzati da MQAI (IBM MQ Administration Interface) per popolare i bag di dati quando vengono creati. Questi elementi dati possono essere elementi utente o di sistema.

Questi elementi utente contengono dati utente, ad esempio attributi di oggetti gestiti. Gli elementi di sistema devono essere utilizzati per un maggiore controllo sui messaggi generati: ad esempio, la creazione di intestazioni di messaggi. Per ulteriori informazioni sugli elementi di sistema, consultare [“Elementi di sistema e MQAI” a pagina 66](#).

## Tipi di elementi dati

Una volta creato un contenitore di dati, è possibile popolarlo con elementi interi o stringa di caratteri. È possibile richiedere informazioni su tutti e tre i tipi di elemento.

L'elemento dati può essere un numero intero o un elemento stringa di caratteri. Di seguito sono riportati i tipi di elementi dati disponibili all'interno di MQAI:

- Intero
- Intero a 64 bit
- Filtro numero intero
- Stringa di caratteri
- Filtro stringa
- Stringa byte
- Filtro stringa di byte
- Maniglia del sacchetto

## Utilizzo di elementi dati

Questi sono i seguenti modi di utilizzare gli elementi dati:

- [“Conteggio elementi dati”](#) a pagina 70.
- [“Eliminazione di elementi dati”](#) a pagina 71.
- [“Aggiunta di elementi dati ai bag con MQAI”](#) a pagina 66.
- [“Filtro e query degli elementi dati”](#) a pagina 67.

### Multi Elementi di sistema e MQAI

Gli elementi di sistema possono essere utilizzati da MQAI ( IBM MQ Administration Interface) per:

- La creazione di intestazione PCF. Le voci di sistema possono controllare l'identificativo del comando PCF, le opzioni di controllo, il numero di sequenza del messaggio e il tipo di comando.
- Conversione dati. Gli elementi del sistema gestiscono l'identificativo della serie di caratteri per gli elementi della stringa di caratteri nel contenitore.

Come tutti gli elementi di dati, gli elementi di sistema sono costituiti da un selettore e un valore. Per informazioni su questi selettori e sulla loro funzione, consultare [Selettori MQAI](#).

Gli elementi di sistema sono univoci. Uno o più elementi di sistema possono essere identificati da un selettore di sistema. C'è solo una ricorrenza di ogni selettore di sistema.

La maggior parte degli elementi del sistema possono essere modificati (consultare [“Modifica delle informazioni all'interno di un contenitore”](#) a pagina 68 ), ma le opzioni di creazione del contenitore non possono essere modificate dall'utente. Non è possibile eliminare elementi di sistema. (Consultare [“Eliminazione di elementi dati”](#) a pagina 71.)

### Multi Aggiunta di elementi dati ai bag con MQAI

Quando un contenitore di dati viene creato utilizzando l'interfaccia di gestione IBM MQ (MQAI), è possibile popolarlo con elementi dati. Questi elementi dati possono essere elementi utente o di sistema.

Per ulteriori informazioni sugli elementi dati, consultare [“Tipi di elementi dati disponibili in MQAI”](#) a pagina 65.

MQAI consente di aggiungere elementi interi, elementi interi a 64 - bit, elementi di filtro interi, elementi stringa di caratteri, filtri stringa, elementi stringa di byte e elementi filtro stringa di byte ai bag, come mostrato in [Figura 5](#) a pagina 66. Gli elementi sono identificati da un selettore. Di solito un selettore identifica solo un elemento, ma non sempre è così. Se un elemento dati con il selettore specificato è già presente nel contenitore, un'istanza aggiuntiva di tale selettore viene aggiunta alla fine del contenitore.



Figura 5. aggiunte degli elementi dati

Aggiungere elementi dati a un contenitore utilizzando le chiamate mqAdd\*:

- Per aggiungere elementi interi, utilizzare la chiamata mqAddInteger come descritto in [mqAddInteger](#)
- Per aggiungere elementi interi a 64 bit, utilizzare la chiamata mqAddInteger64 come descritto in [mqAddInteger64](#)

- Per aggiungere elementi di filtro interi, utilizzare la chiamata `mqAddIntegerFilter` come descritto in [mqAddIntegerFilter](#)
- Per aggiungere elementi stringa di caratteri, utilizzare la chiamata `mqAddString` come descritto in [mqAddString](#)
- Per aggiungere elementi filtro stringa, utilizzare la chiamata `mqAddStringFilter` come descritto in [mqAddStringFilter](#)
- Per aggiungere elementi stringa di byte, utilizzare la chiamata `mqAddByteString` come descritto in [mqAddByteString](#)
- Per aggiungere elementi filtro stringa di byte, utilizzare la chiamata filtro `mqAddByteString` come descritto in [mqAddByteStringFilter](#)

Per ulteriori informazioni sull'aggiunta di elementi dati a un contenitore, consultare [“Elementi di sistema e MQAI”](#) a pagina 66.

### **Multi** Aggiunta di un comando di interrogazione ad un contenitore

La chiamata di interrogazione `mqAdd` viene utilizzata per aggiungere un comando di interrogazione ad un contenitore. La chiamata è specificamente per scopi di amministrazione, quindi può essere utilizzato solo con sacchetti di amministrazione. Consente di specificare i selettori di attributi su cui si desidera eseguire l'interrogazione da IBM MQ.

Per una descrizione completa della chiamata di interrogazione `mqAdd`, consultare [mqAddInquiry](#).

### **Multi** Filtro e query degli elementi dati

Quando si utilizza MQAI per interrogare gli attributi degli oggetti IBM MQ, è possibile controllare i dati restituiti al programma in due modi.

- È possibile **filtrare** i dati restituiti utilizzando le chiamate `mqAddInteger` e `mqAddString`. Questo approccio consente di specificare una coppia *Selector* e *ItemValue*, ad esempio:

```
mqAddInteger(inputbag, MQIA_Q_TYPE, MQQT_LOCAL)
```

Questo esempio specifica che il tipo di coda (*Selector*) deve essere locale (*ItemValue*) e che questa specifica deve essere corrispondente agli attributi dell'oggetto (in questo caso, una coda) che si sta analizzando.

Altri attributi che possono essere filtrati corrispondono ai comandi PCF Inquire \* che possono essere trovati in [“Introduzione a IBM MQ Programmable Command Formats”](#) a pagina 24. Ad esempio, per informazioni sugli attributi di un canale, consultare il comando Interroga canale nella documentazione di questo prodotto. I "Parametri obbligatori" e i "Parametri facoltativi" del comando Inquire Channel identificano i selettori che è possibile utilizzare per il filtro.

- È possibile **interrogare** particolari attributi di un oggetto utilizzando la chiamata di interrogazione `mqAdd`. Specifica il selettore a cui si è interessati. Se non si specifica il selettore, vengono restituiti tutti gli attributi dell'oggetto.

Di seguito viene riportato un esempio di filtro ed esecuzione di query degli attributi di una coda:

```
/* Request information about all queues */
mqAddString(adminbag, MQCA_Q_NAME, "*")

/* Filter attributes so that local queues only are returned */
mqAddInteger(adminbag, MQIA_Q_TYPE, MQQT_LOCAL)

/* Query the names and current depths of the local queues */
mqAddInquiry(adminbag, MQCA_Q_NAME)
mqAddInquiry(adminbag, MQIA_CURRENT_Q_DEPTH)

/* Send inquiry to the command server and wait for reply */
mqExecute(MQCMD_INQUIRE_Q, ...)
```

**Multi** Richiesta all'interno dei bag di dati

È possibile richiedere informazioni su:

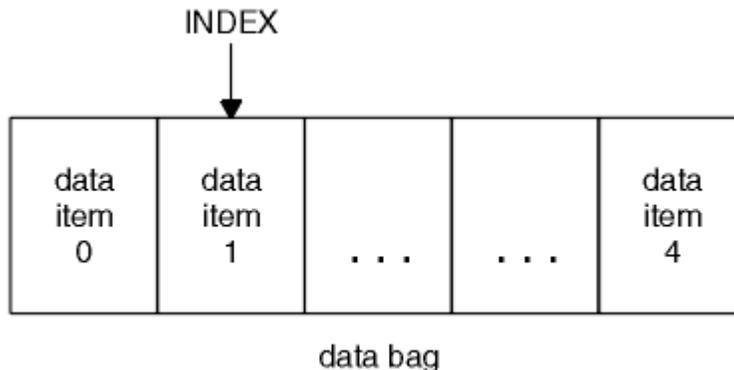
- Il valore di un elemento intero che utilizza la chiamata numero intero mqInquire. Vedere [mqInquireInteger](#).
- Il valore di un elemento intero a 64 bit utilizzando la chiamata mqInquireInteger64 . Vedere [mqInquireInteger64](#).
- Il valore di un elemento filtro intero che utilizza la chiamata mqInquireIntegerFilter . Vedere [mqInquireIntegerFilter](#).
- Il valore di un elemento stringa di caratteri che utilizza la chiamata stringa mqInquire. Vedere [mqInquiremqInquire](#).
- Il valore di un elemento filtro stringa utilizzando la chiamata mqInquireStringFilter . Vedere [mqInquireStringFilter](#).
- Il valore di un elemento stringa di byte utilizzando la chiamata mqInquireByteString . Vedere [mqInquireByteString](#).
- Il valore di un elemento filtro stringa di byte utilizzando la chiamata Filtro mqInquireByteString. Consultare [mqInquireByteStringFilter](#).
- Il valore di un handle di borsa utilizzando la chiamata di borsa mqInquire. Vedere [mqInquireBag](#).

È anche possibile richiedere informazioni sul tipo (integer, 64 - bit integer, integer filter, character string, string filter, byte string, byte string filter o bag handle) di un elemento specifico utilizzando la chiamata mqInquireItemInfo . Vedere [mqInquireItemInfo](#).

**Multi** Modifica delle informazioni all'interno di un contenitore

MQAI consente di modificare le informazioni all'interno di un contenitore utilizzando le chiamate mqSet \*. È possibile effettuare quanto segue.

1. Modificare gli elementi dati all'interno di un contenitore. L'indice consente di sostituire una singola istanza di un parametro identificando la ricorrenza dell'elemento da modificare (consultare [Figura 6 a pagina 68](#)).



*Figura 6. Modifica di un singolo elemento dati*

2. Eliminare tutte le ricorrenze esistenti del selettore specificato e aggiungere una nuova ricorrenza alla fine del contenitore. (Consultare [Figura 7 a pagina 69](#).) Un valore di indice speciale consente di sostituire **tutte le** istanze di un parametro.

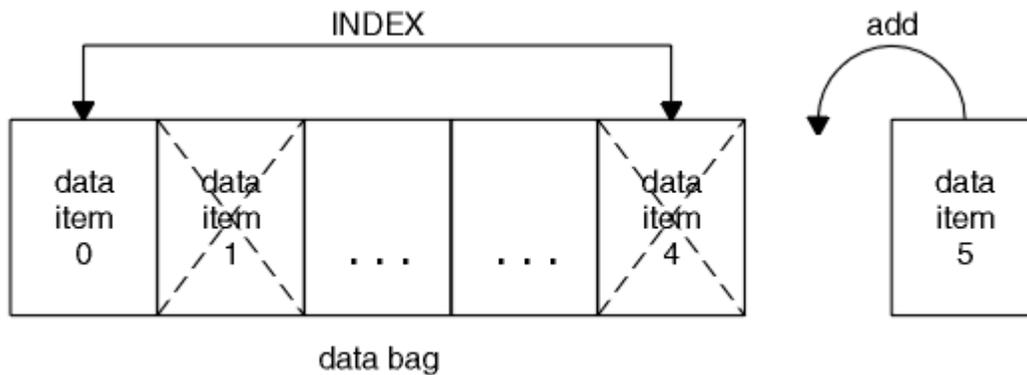


Figura 7. Modifica di tutti gli elementi dati

**Nota:** L'indice conserva l'ordine di inserimento all'interno del contenitore ma può influire sugli indici di altri elementi dati.

La chiamata `mqSetInteger` consente di modificare gli elementi interi in un contenitore. La chiamata `mqSetInteger64` consente di modificare elementi interi a 64 bit. La chiamata `mqSetIntegerFilter` consente di modificare gli elementi del filtro integer. La chiamata `mqSetString` consente di modificare gli elementi stringa di caratteri. La chiamata `mqSetStringFilter` consente di modificare gli elementi filtro stringa. La chiamata `mqSetByteString` consente di modificare gli elementi stringa di byte. La chiamata `mqSetByteStringFilter` consente di modificare gli elementi filtro della stringa di byte. In alternativa, è possibile utilizzare queste chiamate per eliminare tutte le ricorrenze esistenti del selettore specificato e aggiungere una nuova ricorrenza alla fine del contenitore. L'elemento dati può essere un elemento utente o un elemento di sistema.

Per una descrizione completa di queste chiamate, consultare:

- [mqSetInteger](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [MQSetStringa](#)
- [Filtro mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringFilter](#)

**Multi** *Cancellazione di un contenitore utilizzando la chiamata `mqClearBag`*

La chiamata `mqClearBag` rimuove tutti gli elementi utente da un contenitore utente e reimposta gli elementi di sistema sui valori iniziali. Vengono eliminati anche i sacchetti di sistema contenuti all'interno del sacchetto.

Per una descrizione completa della chiamata `mqClearBag`, vedere [mqClearBag](#).

**Multi** *Troncamento di un sacchetto utilizzando la chiamata `mqTruncateBag`*

La chiamata `mqTruncateBag` riduce il numero di elementi utente in una borsa utente eliminando gli elementi dalla fine della borsa, a partire dall'elemento aggiunto più di recente. Ad esempio, può essere utilizzato quando si utilizzano le stesse informazioni di intestazione per generare più di un messaggio.

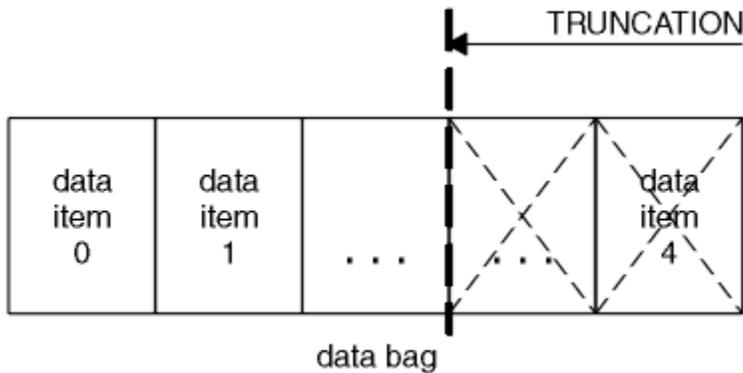


Figura 8. Troncamento di un sacchetto

Per una descrizione completa della chiamata `mqTruncateBag`, consultare [mqTruncateBag](#).

### Multi Conversione di buste e buffer

Per inviare i dati tra le applicazioni, innanzitutto i dati del messaggio vengono inseriti in un contenitore. Quindi, i dati nel contenitore vengono convertiti in un messaggio PCF utilizzando la chiamata `mqBagToBuffer`. Il messaggio PCF viene inviato alla coda richiesta utilizzando la chiamata `MQPUT`. Ciò è mostrato in Figura 9 a pagina 70. Per una descrizione completa della chiamata `mqBagToBuffer`, consultare [mqBagToBuffer](#).

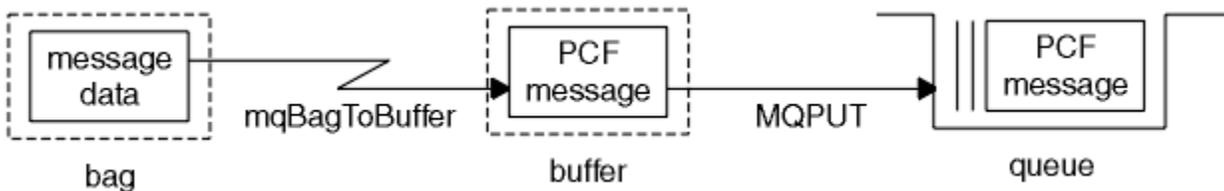


Figura 9. Conversione di sacchetti in messaggi PCF

Per ricevere i dati, il messaggio viene ricevuto in un buffer utilizzando la chiamata `MQGET`. I dati nel buffer vengono quindi convertiti in un contenitore utilizzando la chiamata `mqBufferToBag`, purché il buffer contenga un messaggio PCF valido. Ciò è mostrato in Figura 10 a pagina 70. Per una descrizione completa della chiamata `mqBufferToBag`, consultare [mqBufferToBag](#).

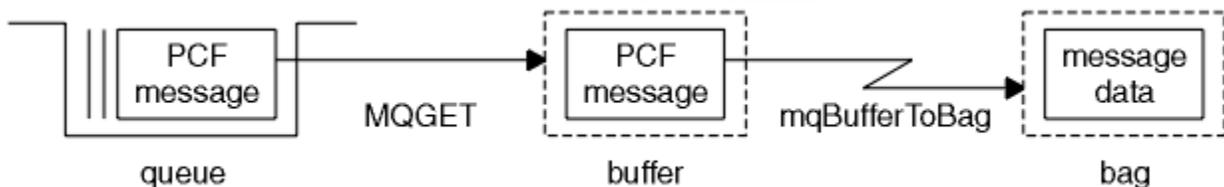


Figura 10. Conversione dei messaggi PCF in formato bag

### Multi Conteggio elementi dati

La chiamata `mqCountItems` conta il numero di elementi utente, elementi di sistema o entrambi, memorizzati in un contenitore di dati e restituisce questo numero. Ad esempio, `mqCountItems( Bag, 7, ... )`, restituisce il numero di elementi nella borsa con un selettore di 7. Può contare gli elementi per singolo selettore, per selettori utente, per selettori di sistema o per tutti i selettori.

**Nota:** Questa chiamata conta il numero di elementi dati, non il numero di selettori univoci nel contenitore. Un selettore può verificarsi più volte, quindi potrebbero esserci meno selettori univoci nel contenitore rispetto agli elementi di dati.

Per una descrizione completa della chiamata `mqCountItems`, consultare [mqCountItems](#).

**Multi** *Eliminazione di elementi dati*

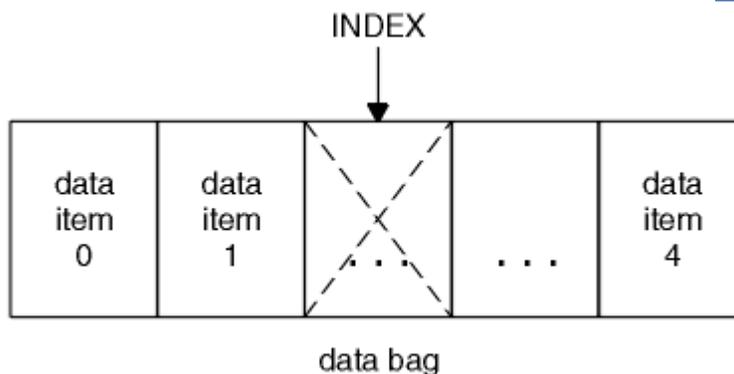
È possibile eliminare gli articoli dalle borse in diversi modi. È possibile effettuare quanto segue.

- Rimuovere uno o più elementi utente da un contenitore. Per informazioni dettagliate, consultare [“Eliminazione di elementi dati da un contenitore utilizzando la chiamata mqDeleteItem”](#) a pagina 71.
- Eliminare tutti gli elementi utente da un contenitore, ossia cancellare un contenitore. Per informazioni dettagliate, consultare [“Cancellazione di un contenitore utilizzando la chiamata mqClearBag”](#) a pagina 69.
- Eliminare gli elementi utente dalla fine di un contenitore, ovvero troncare un contenitore. Per informazioni dettagliate, consultare [“Troncamento di un sacchetto utilizzando la chiamata mqTruncateBag”](#) a pagina 69.

**Multi** *Eliminazione di elementi dati da un contenitore utilizzando la chiamata mqDeleteItem*

La chiamata mqDeleteItem rimuove uno o più elementi utente da un contenitore. L'indice viene utilizzato per eliminare:

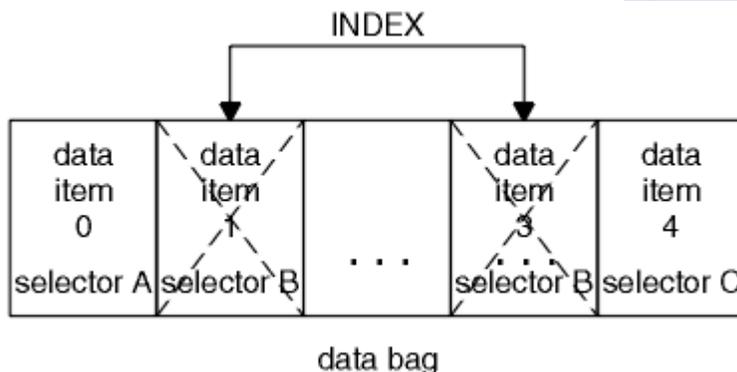
1. Una singola ricorrenza del selettore specificato. (Consultare [Figura 11](#) a pagina 71.)



*Figura 11. Eliminazione di un singolo elemento dati*

oppure

2. Tutte le ricorrenze del selettore specificato. (Consultare [Figura 12](#) a pagina 71.)



*Figura 12. Eliminazione di tutti gli elementi dati*

**Nota:** L'indice conserva l'ordine di inserimento all'interno del contenitore ma può influire sugli indici di altri elementi dati. Ad esempio, la chiamata mqDeleteItem non conserva i valori di indice degli elementi dati che seguono l'elemento eliminato perché gli indici sono riorganizzati per colmare il divario che rimane dall'elemento eliminato.

Per una descrizione completa della chiamata mqDeleteItem, consultare [mqDeleteItem](#).

## Invio dei comandi di gestione al server dei comandi qm utilizzando la chiamata mqExecute

Quando una serie di dati è stata creata e popolata, è possibile inviare un messaggio di comando di gestione al server dei comandi di un gestore code utilizzando la chiamata mqExecute . Gestisce lo scambio con il server dei comandi e restituisce le risposte in un contenitore.

Una volta creato e popolato il contenitore dati, è possibile inviare un messaggio di comando di gestione al server dei comandi di un gestore code. Il modo più semplice per eseguire questa operazione è utilizzando la chiamata mqExecute . La chiamata mqExecute invia un messaggio di comando di gestione come messaggio non persistente e attende eventuali risposte. Le risposte vengono restituite in un contenitore di risposte. Queste potrebbero contenere informazioni relative agli attributi relativi a diversi oggetti IBM MQ o una serie di messaggi di risposta di errore PCF, ad esempio. Pertanto, il contenitore della risposta potrebbe contenere solo un codice di ritorno o potrebbe contenere *contenitori nidificati*.

I messaggi di risposta vengono inseriti nei bag di sistema creati dal sistema. Ad esempio, per richieste relative ai nomi degli oggetti, viene creata una serie di sistemi per contenere tali nomi oggetto e la serie viene inserita nella serie utente. Le maniglie di queste borse vengono quindi inserite nel sacchetto di risposta e al sacchetto nidificato può accedere il selettore MQHA\_BAG\_HANDLE. Il contenitore di sistema rimane in memoria, se non viene eliminato, fino a quando il contenitore di risposta non viene eliminato.

Il concetto di *nidificazione* viene mostrato in [Figura 13 a pagina 72](#).

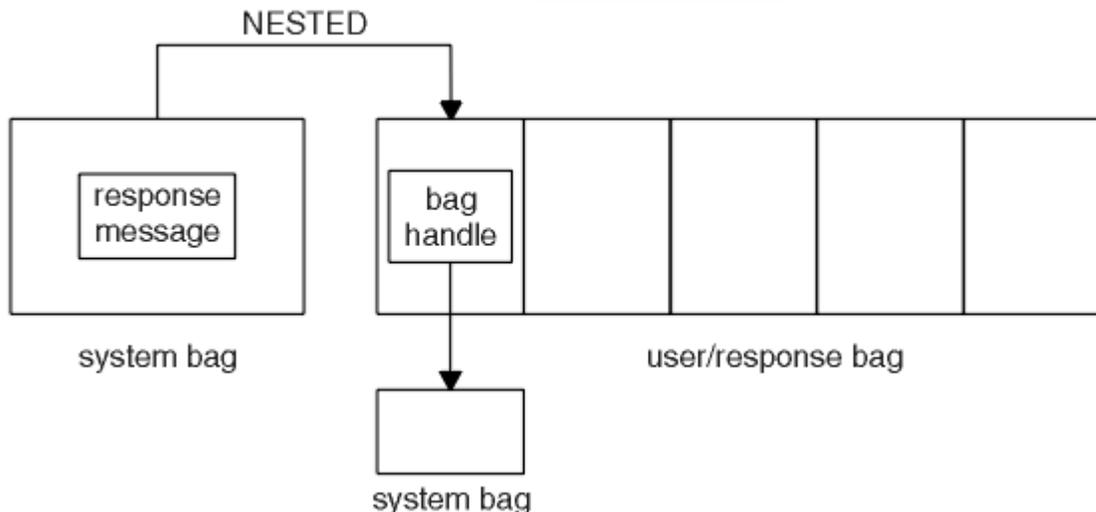


Figura 13. Nidificazione

Come input per la chiamata mqExecute , è necessario fornire:

- Un handle di connessione MQI.
- Il comando da eseguire. Deve essere uno dei valori MQCMD\_ \*.

**Nota:** Se questo valore non viene riconosciuto da MQAI, viene comunque accettato. Tuttavia, se la chiamata mqAddInquiry è stata utilizzata per inserire i valori nel contenitore, questo parametro deve essere un comando INQUIRE riconosciuto da MQAI. Ovvero, il parametro deve essere nel formato MQCMD\_INQUIRE\_ \*.

- Facoltativamente, un handle del contenitore contenente le opzioni che controllano l'elaborazione della chiamata. Questo è anche il punto in cui è possibile specificare il tempo massimo in millisecondi che MQAI deve attendere per ciascun messaggio di risposta.
- Un handle del contenitore di gestione che contiene i dettagli del comando di gestione da immettere.
- Un handle del contenitore risposte che riceve i messaggi di risposta.

I seguenti handle sono facoltativi:

- Un handle di oggetto della coda in cui deve essere inserito il comando di gestione.

Se non viene specificato alcun handle di oggetto, il comando di gestione viene posizionato su SYSTEM.ADMIN.COMMAND.QUEUE appartenente al Gestore code attualmente connesso. Questa è l'opzione predefinita.

- Una gestione oggetto della coda in cui devono essere posizionati i messaggi di risposta.

È possibile scegliere di inserire i messaggi di risposta in una coda dinamica creata automaticamente da MQAI. La coda creata esiste solo per la durata della chiamata e viene eliminata dall'MQAI all'uscita dalla chiamata mqExecute .

Per esempi di utilizzo della chiamata mqExecute , consultare [Codice di esempio](#)

## Amministrazione mediante REST API

---

È possibile utilizzare amministrative REST API per amministrare gli oggetti IBM MQ , come i gestori code e le code, gli agent e i trasferimenti Managed File Transfer . Le informazioni vengono inviate e ricevute da amministrative REST API in formato JSON. Queste API RESTful possono aiutarti a integrare la gestione IBM MQ negli strumenti DevOps e di automazione.

### Prima di iniziare

Per informazioni di riferimento sulle risorse REST disponibili, consultare [Il riferimento amministrative REST API](#).

### Procedura

- [“Introduzione a amministrative REST API” a pagina 73](#)
- [“Utilizzo di amministrative REST API” a pagina 77](#)
- [“Gestione remota mediante REST API” a pagina 78](#)
- [“REST API data/ora” a pagina 82](#)
- [“REST API gestione degli errori” a pagina 83](#)
- [“Rilevamento REST API” a pagina 85](#)
- [“REST API supporto lingua nazionale” a pagina 86](#)

## Introduzione a amministrative REST API

Inizia rapidamente con amministrative REST API e prova alcune richieste di esempio utilizzando cURL per creare, aggiornare, visualizzare ed eliminare una coda.

### Prima di iniziare

Per iniziare a utilizzare amministrative REST API, gli esempi in questa attività hanno i seguenti requisiti:

- Gli esempi utilizzano cURL per effettuare richieste REST per visualizzare informazioni sui gestori code sul sistema e per creare una coda, aggiornare, visualizzare ed eliminare una coda. Pertanto, per completare questa attività è necessario che cURL sia installato sul sistema.
- Per completare questa attività, è necessario essere un utente con particolari privilegi in modo da poter utilizzare il comando **dspmweb**:

–  **z/OS** Su z/OS, è necessario disporre dell'autorità per eseguire il comando **dspmweb** e l'accesso in scrittura al file mqwebuser.xml.

–  **Multi** Su tutti gli altri sistemi operativi, è necessario essere un [utente privilegiato](#).

 **IBM i** Su IBM i, i comandi devono essere in esecuzione in QSHELL.

## Procedura

1. Se il server mqweb non è già configurato per l'utilizzo da parte di amministrative REST API, amministrative REST API per MFT, messaging REST APIo IBM MQ Console, configurare il server mqweb.

Per ulteriori informazioni sulla creazione di una configurazione di base per il server mqweb con un registro di base, consultare [Configurazione di base per il server mqweb](#).

2.  Su z/OS, impostare la variabile di ambiente WLP\_USER\_DIR in modo da poter utilizzare il comando **dspmweb**. Impostare la variabile in modo che punti alla configurazione del server mqweb immettendo il seguente comando:

```
export WLP_USER_DIR=WLP_user_directory
```

dove *WLP\_user\_directory* è il nome della directory passata a *crtmqweb*. Ad esempio:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Per ulteriori informazioni, consultare [Creazione del server mqweb](#).

3. Determinare l'URL REST API immettendo il seguente comando:

```
dspmweb status
```

Gli esempi riportati di seguito presuppongono che l'URL REST API sia l'URL predefinito `https://localhost:9443/ibmmq/rest/v1/`. Se l'URL è diverso da quello predefinito, sostituirlo nella seguente procedura.

4. Prova una richiesta GET nella risorsa `qmgr` utilizzando l'autenticazione di base con l'utente `mqadmin`:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/qmgr -X GET -u mqadmin:mqadmin
```

5. Creare, visualizzare, modificare ed eliminare una coda utilizzando la risorsa `mqsc`:

Questo esempio utilizza un gestore code QM1. Creare un gestore code con lo stesso nome o sostituire un gestore code esistente sul proprio sistema.

- a) Effettuare una richiesta POST sulla risorsa `mqsc` per creare la coda locale:

Nel corpo della richiesta, il nome della nuova coda è impostato su Q1. Viene utilizzata l'autenticazione di base e viene impostata un'intestazione HTTP `ibm-mq-rest-csrf-token` con un valore arbitrario nella richiesta REST cURL. Questa intestazione aggiuntiva è richiesta per le richieste POST, PATCH e DELETE:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data "{\"type\": \"runCommandJSON\", \"command\": \"define\", \"qualifier\": \"qlocal\", \"name\": \"Q1\"}"
```

- b) Effettuare una richiesta POST sulla risorsa `mqsc` per visualizzare la coda locale creata nel passo ["5.a"](#) a pagina 74:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data "{\"type\": \"runCommandJSON\", \"command\": \"display\", \"qualifier\": \"qlocal\", \"name\": \"Q1\"}"
```

- c) Effettuare una richiesta POST sulla risorsa `mqsc` per aggiornare la descrizione della coda:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data "{\"type\": \"runCommandJSON\", \"command\": \"alter\", \"qualifier\": \"qlocal\", \"name\": \"Q1\", \"parameters\": {\"descr\": \"new description\" }}"
```

- d) Effettuare una richiesta POST sulla risorsa mqsc per visualizzare la nuova descrizione della coda. Specificare l'attributo **responseParameters** nel corpo della richiesta in modo che la risposta includa il campo della descrizione:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1", "responseParameters": [{"descr"}]}'
```

- e) Effettuare una richiesta POST sulla risorsa di mqsc per eliminare la coda:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "delete", "qualifier": "qlocal", "name": "Q1"}'
```

- f) Effettuare una richiesta POST sulla risorsa mqsc per provare che la coda è stata eliminata:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

## Operazioni successive

- Gli esempi utilizzano l'autenticazione di base per proteggere la richiesta. È possibile utilizzare invece l'autenticazione basata su token o sul client. Per ulteriori informazioni, consultare [Utilizzo dell'autenticazione del certificato client con la REST API](#) e [Utilizzo dell'autenticazione basata su token con la REST API](#).
- Ulteriori informazioni sull'utilizzo di amministrative REST API e sulla creazione di URL con parametri di query: [“Utilizzo di amministrative REST API” a pagina 77](#).
- Sfogliare le informazioni di riferimento per le risorse amministrative REST API disponibili e tutti i parametri di query facoltativi disponibili: [administrative REST API reference](#).
- Informazioni su come utilizzare amministrative REST API per gestire oggetti IBM MQ su sistemi remoti: [“Gestione remota mediante REST API” a pagina 78](#).
- Informazioni sull'utilizzo di amministrative REST API con MFT: [“Introduzione a REST API per MFT” a pagina 75](#).
- Rilevare messaging REST API, un'interfaccia RESTful per la messaggistica IBM MQ : [Messaggistica utilizzando REST API](#).
- Rilevare IBM MQ Console, una GUI basata su browser: [“Gestione mediante una console Web” a pagina 90](#).

## Introduzione a REST API per MFT

Iniziare rapidamente con amministrative REST API per Managed File Transfer e provare alcune richieste di esempio per visualizzare lo stato dell'agent MFT e per visualizzare un elenco di trasferimenti.

### Prima di iniziare

- Gli esempi utilizzano cURL per inviare le richieste REST per visualizzare un elenco di trasferimenti e visualizzare lo stato dell'agent MFT . Pertanto, per completare questa attività è necessario che cURL sia installato sul sistema.
- Per completare questa attività, è necessario essere un utente con particolari privilegi in modo da poter utilizzare il comando **dspmweb**:

-  Su z/OS, è necessario disporre dell'autorità per eseguire il comando **dspmweb** e l'accesso in scrittura al file `mqwebuser.xml`.
-  Su tutti gli altri sistemi operativi, è necessario essere un utente privilegiato.

## Procedura

1. Verificare che il server mqweb sia configurato per administrative REST API per MFT:

- Se il server mqweb non è già configurato per l'utilizzo da parte di administrative REST API, administrative REST API per MFT, messaging REST API o IBM MQ Console, configurare il server mqweb. Per ulteriori informazioni sulla creazione di una configurazione di base per il server mqweb con un registro di base, consultare [Configurazione di base per il server mqweb](#).
- Se il server mqweb è configurato, assicurarsi che il passo 8 di [Configurazione di base per il server mqweb](#) sia stato completato per abilitare administrative REST API per MFT.

2. 

Su z/OS, impostare la variabile di ambiente WLP\_USER\_DIR in modo da poter utilizzare il comando **dspmweb**. Impostare la variabile in modo che punti alla configurazione del server mqweb immettendo il seguente comando:

```
export WLP_USER_DIR=WLP_user_directory
```

dove *WLP\_user\_directory* è il nome della directory passata a `crtmqweb`. Ad esempio:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Per ulteriori informazioni, consultare [Creazione del server mqweb](#).

3. Determinare l'URL REST API immettendo il seguente comando:

```
dspmweb status
```

Gli esempi riportati di seguito presuppongono che l'URL REST API sia l'URL predefinito `https://localhost:9443/ibmmq/rest/v1/`. Se l'URL è diverso da quello predefinito, sostituirlo nella seguente procedura.

4. Eseguire una richiesta GET sulla risorsa `agent` per restituire i dettagli di base su tutti gli agent, inclusi il nome, il tipo e lo stato:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/agent/ -X GET -u mftadmin:mftadmin
```

5. Creare alcuni trasferimenti da visualizzare utilizzando il comando **fteCreateTransfer**.

Il server mqweb memorizza nella cache le informazioni sui trasferimenti e restituisce tali informazioni quando viene effettuata una richiesta. Questa cache viene reimpostata quando il server mqweb viene riavviato. È possibile verificare se il server è stato riavviato visualizzando i file `console.log` e `messages.log` oppure su z/OS, esaminando l'output dell'attività avviata.

6. Effettuare una richiesta GET sulla risorsa `transfer` per restituire i dettagli di un massimo di quattro trasferimenti effettuati da quando è stato avviato il server mqWeb:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/transfer?limit=4 -X GET -u mftadmin:mftadmin
```

## Operazioni successive

- Gli esempi utilizzano l'autenticazione di base per proteggere la richiesta. È possibile utilizzare invece l'autenticazione basata su token o sul client. Per ulteriori informazioni, consultare [Utilizzo dell'autenticazione basata sul token con REST API](#) e [Utilizzo dell'autenticazione del certificato client con REST API e IBM MQ Console](#).
- Ulteriori informazioni sull'utilizzo di administrative REST API e sulla creazione di URL con parametri di query: ["Utilizzo di administrative REST API" a pagina 77](#).
- Sfogliare le informazioni di riferimento per le risorse administrative REST API per MFT disponibili e tutti i parametri di query facoltativi disponibili: [administrative REST API reference](#).

- Rilevare messaging REST API, un'interfaccia RESTful per la messaggistica IBM MQ : [Messaggistica utilizzando REST API](#).
- Rilevare IBM MQ Console, una GUI basata su browser: [“Gestione mediante una console Web” a pagina 90](#).

## Utilizzo di administrative REST API

Quando si utilizza administrative REST API, si richiamano i metodi HTTP sugli URL che rappresentano i vari oggetti IBM MQ , come i gestori code o le code. Il metodo HTTP, ad esempio POST, rappresenta il tipo di azione da eseguire sull'oggetto rappresentato dall'URL. Ulteriori informazioni sull'azione potrebbero essere fornite in JSON come parte del payload del metodo HTTP o codificate nei parametri della query. Le informazioni sul risultato dell'esecuzione dell'azione potrebbero essere restituite come corpo della risposta HTTP.

### Prima di iniziare

Considerare quanto segue prima di utilizzare administrative REST API:

- È necessario autenticarsi con il server mqweb per utilizzare administrative REST API. È possibile eseguire l'autenticazione utilizzando l'autenticazione di base HTTP, l'autenticazione del certificato client o l'autenticazione basata su token. Per ulteriori informazioni su come utilizzare questi metodi di autenticazione, consultare [IBM MQ Console and REST API security](#).
- REST API è sensibile al maiuscolo / minuscolo. Ad esempio, un HTTP GET sul seguente URL non visualizza le informazioni se il gestore code è denominato qmgr1.

```
/ibmmq/rest/v1/admin/qmgr/QMGR1
```

- Non tutti i caratteri che possono essere utilizzati nei nomi oggetto IBM MQ possono essere codificati direttamente in un URL. Per codificare correttamente questi caratteri, è necessario utilizzare la codifica URL appropriata:
  - Una barra, /, deve essere codificata come %2F.
  - Un segno di percentuale, %, deve essere codificato come %25.
- A causa del funzionamento di alcuni browser, non denominare gli oggetti utilizzando solo i caratteri punto o barra.

### Informazioni su questa attività

Quando si usa REST API per eseguire un'azione su un oggetto, è necessario prima costruire un URL per rappresentare tale oggetto. Ogni URL inizia con un prefisso, che descrive a quale nome host e porta inviare la richiesta. Il resto dell'URL descrive un oggetto particolare o una serie di oggetti, noti come una risorsa.

L'azione che deve essere eseguita sulla risorsa definisce se l'URL necessita o meno di parametri di query. Definisce anche il metodo HTTP utilizzato e se le informazioni aggiuntive vengono inviate all'URL, o restituite da esso, in formato JSON. Le informazioni aggiuntive potrebbero far parte della richiesta HTTP o essere restituite come parte della risposta HTTP.

Dopo aver creato l'URL e creato un payload JSON facoltativo per l'invio nella richiesta HTTP, puoi inviare la richiesta HTTP a IBM MQ. È possibile inviare la richiesta utilizzando l'implementazione HTTP integrata nel linguaggio di programmazione scelto. Puoi anche inviare le richieste utilizzando strumenti della riga di comando come cURL, un browser web o un componente aggiuntivo del browser web.

**Importante:** È necessario, come minimo, eseguire le operazioni [“1.a” a pagina 77](#) e [“1.b” a pagina 78](#).

### Procedura

1. Creare l'URL:
  - a) Determinare l'URL del prefisso immettendo il seguente comando:

```
dspmweb status
```

L'URL che vuoi utilizzare include la frase `/ibmmq/rest/`.

b) Aggiungere la risorsa al percorso URL.

Sono disponibili le seguenti risorse IBM MQ :

- [/admin/installazione](#)
- [/admin/qmgr](#)
- [/admin/coda](#)
- [/admin/sottoscrizione](#)
- [/admin/canale](#)
- [/action/qmgr/{qmgrname}/mqsc](#)

Sono disponibili le seguenti risorse Managed File Transfer :

- [/admin/agent](#)
- [/admin/trasferimento](#)
- [/admin/monitor](#)

Ad esempio, per interagire con i gestori code, aggiungere `/qmgr` all'URL del prefisso per creare il seguente URL:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr
```

c) Opzionale: Aggiungere ulteriori segmenti di percorso facoltativi all'URL.

Nelle informazioni di riferimento per ciascun tipo di oggetto, i segmenti facoltativi possono essere identificati nell'URL dalle parentesi graffe che lo circondano `{ }`.

Ad esempio, aggiungere il nome gestore code QM1 all'URL per creare il seguente URL:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1
```

d) Opzionale: Aggiungere un parametro di query facoltativo all'URL.

Aggiungere un punto interrogativo, `?`, nome della variabile, segno uguale `=` e un valore o un elenco di valori per l'URL.

Ad esempio, per richiedere tutti gli attributi del gestore code QM1, creare il seguente URL:

```
https://localhost:9443/ibmmq/rest/v2/admin/qmgr/QM1?attributes=*
```

e) Aggiungere ulteriori parametri di query facoltativi all'URL.

Aggiungere una e commerciale, `&`, all'URL e ripetere il [passo d.](#)

2. Richiamare il metodo HTTP pertinente sull'URL. Specificare qualsiasi payload JSON facoltativo e fornire le credenziali di sicurezza appropriate per l'autenticazione. Ad esempio:

- Utilizzare l'implementazione HTTP/REST del linguaggio di programmazione scelto.
- Utilizza uno strumento come un componente aggiuntivo del browser del client REST o cURL.

## Gestione remota mediante REST API

È possibile utilizzare REST API per gestire i gestori code remoti e gli oggetti IBM MQ associati a tali gestori code. Questa gestione remota include gestori code che si trovano nello stesso sistema, ma che non si trovano nella stessa installazione IBM MQ del server mqweb. Pertanto, è possibile utilizzare REST API per gestire l'intera rete IBM MQ con una sola installazione che esegue il server mqweb. Per gestire i gestori code remoti, è necessario configurare il gateway administrative REST API in modo tale che almeno un gestore code nella stessa installazione del server mqweb agisca come gestore code del gateway.

Quindi, è possibile specificare il gestore code remoto nell'URL della risorsa REST API per eseguire l'azione amministrativa specificata.

## Prima di iniziare

È possibile impedire la gestione remota disabilitando il gateway amministrative REST API . Per ulteriori informazioni, vedi [Configurazione del gateway amministrative REST API](#).

Per utilizzare il gateway amministrative REST API , devono essere soddisfatte le seguenti condizioni:

- Il server mqweb deve essere configurato e avviato. Per ulteriori informazioni sulla configurazione e l'avvio del server mqweb, consultare [“Introduzione a amministrative REST API”](#) a pagina 73.
- Il gestore code che si desidera configurare come gestore code del gateway deve trovarsi nella stessa installazione del server mqweb.
- Il gestore code remoto che si desidera gestire deve essere IBM MQ 8.0 o successivo.
- È necessario assicurarsi che gli attributi specificati nella richiesta siano validi per il sistema a cui si sta inviando la richiesta. Ad esempio, se il gestore code del gateway si trova su Windows e il gestore code remoto si trova su z/OS, non è possibile richiedere che l'attributo `dataCollection.statistics` venga restituito per una richiesta HTTP GET sulla risorsa `queue` .
- È necessario assicurarsi che gli attributi specificati nella richiesta siano validi per il livello di IBM MQ a cui si sta inviando la richiesta. Ad esempio, se il gestore code remoto sta eseguendo IBM MQ 8.0, non è possibile richiedere che l'attributo `extended.enableMediaImageOperations` venga restituito per una richiesta HTTP GET sulla risorsa `queue` .
- È necessario utilizzare una delle seguenti risorse REST supportate:
  - `/queue`
  - `/subscription`
  - `/channel`
  - `/mqsc`
  - `/qmgr`

La risorsa `/qmgr` restituisce solo un sottoinsieme di attributi quando si interroga un gestore code remoto: `name`, `status.started`, `status.channelInitiatorState`, `status.ldapConnectionState`, `status.connectionCount` e `status.publishSubscribeState`.

## Informazioni su questa attività

Per utilizzare il gateway amministrative REST API per gestire i gestori code remoti, è necessario preparare i gestori code per la gestione remota. Ovvero, è necessario configurare le code di trasmissione, i listener e i canali mittente e destinatario tra il gestore code del gateway e il gestore code remoto. È quindi possibile inviare una richiesta REST al gestore code remoto specificando il gestore code nell'URL della risorsa. Il gestore code del gateway viene specificato utilizzando il comando **setmqweb** per impostare l'attributo `mqRestGatewayQmgr` sul nome del gestore code del gateway oppure inviando il nome del gestore code del gateway in un'intestazione inviata con la richiesta. La richiesta viene inviata tramite il gestore code del gateway al gestore code remoto. La risposta viene restituita con un'intestazione che indica il gestore code utilizzato come gestore code del gateway.

## Procedura

1. Configurare le comunicazioni tra il gestore code del gateway e i gestori code remoti che si desidera amministrare. Questi passi di configurazione sono gli stessi necessari per configurare la gestione remota da parte di `runmqsc` e PCF.

Per ulteriori informazioni su questi passi, consultare [“Configurazione dei gestori code per la gestione remota”](#) a pagina 188.

2. Configurare la sicurezza sui gestori code remoti:

- a) Verificare che gli ID utente pertinenti esistano sul sistema su cui è in esecuzione il gestore code remoto. L'ID utente che deve esistere sul sistema remoto dipende dal ruolo dell'utente REST API :
    - Se l'utente REST API si trova nel gruppo RO MQWebAdmin o MQWebAdmin, l'ID utente che ha avviato il server mqweb deve esistere sul sistema remoto. Su IBM MQ Appliance, l'utente che avvia il server mqweb è mqsystem.
    - Se l'utente REST API fa parte del gruppo MQWebUser , l'ID utente REST API deve esistere sul sistema remoto.
  - b) Verificare che agli ID utente pertinenti siano concessi i livelli di autorizzazione necessari per accedere alle risorse REST API appropriate sul gestore code remoto:
    - Autorizzazione a inserire messaggi in SYSTEM.ADMIN.COMMAND.QUEUE.
    - Autorizzazione a inserire messaggi in SYSTEM.REST.REPLY.QUEUE.
    - Autorizzazione ad accedere alle code di trasmissione definite per la gestione remota.
    - Autorizzazione a visualizzare gli attributi del gestore code.
    - Autorizzazione per eseguire le richieste REST. Per ulteriori informazioni, consultare la sezione Requisiti di sicurezza degli argomenti di riferimento delle risorse REST API.
3. Configurare quale gestore code locale viene utilizzato come gateway. È possibile configurare un gestore code del gateway predefinito, specificare il gestore code del gateway in un'intestazione HTTP o utilizzare una combinazione di entrambi gli approcci:
- Configurare un gestore code gateway predefinito utilizzando il comando **setmqweb** :

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

dove *qmgrName* è il nome del gestore code del gateway.

Questo gestore code gateway viene utilizzato quando si verificano entrambe le seguenti condizioni:

- Un gestore code non è specificato nell'intestazione `ibm-mq-rest-gateway-qmgr` di una richiesta REST.
  - Il gestore code specificato nell'URL della risorsa REST API non è un gestore code locale.
- Configurare il gestore code gateway su ogni richiesta REST impostando l'intestazione HTTP `ibm-mq-rest-gateway-qmgr` sul nome del gestore code gateway.
4. Includere il nome del gestore code remoto che si desidera gestire nell'URL della risorsa.

Ad esempio, per ottenere un elenco di code dal gestore code remoto `remoteQM`, utilizzare il seguente URL:

```
https://localhost:9443/ibmmq/rest/v1/admin/qmgr/remoteQM/queue
```

## Risultati

Viene restituita un'intestazione `ibm-mq-rest-gateway-qmgr` con la risposta REST. Questa intestazione specifica quale gestore code è stato utilizzato come gestore code del gateway.

In caso di problemi con l'utilizzo di amministrative REST API per gestire i gestori code remoti:

- Verificare che il gestore code remoto sia in esecuzione.
- Controllare che il server dei comandi sia in esecuzione sul sistema remoto.
- Verificare che l'intervallo di disconnessione del canale non sia scaduto. Ad esempio, se un canale è stato avviato ma è stato chiuso dopo un certo periodo di tempo. Questo è particolarmente importante se si avviano i canali manualmente.

## Esempio

Nel seguente esempio, sono presenti tre installazioni IBM MQ su due macchine. Su Machine 1, sono presenti un Installation 1 e un Installation 2. Su Machine 2, è presente un Installation 3. Un server mqweb è configurato per Installation 1. C'è un singolo gestore code in ogni installazione

e questi gestori code sono configurati per la gestione remota. Ciò significa che i seguenti listener, canali e code sono configurati e avviati:

- Sul gestore code QM1, in Installation 1, su Machine 1:
  - Canale mittente QM1.to.QM2
  - Canale ricevente QM2.to.QM1
  - Canale mittente QM1.to.QM3
  - Canale ricevente QM3.to.QM1
  - Coda di trasmissione QM2
  - Coda di trasmissione QM3
  - Un listener configurato sulla porta 1414
- Sul gestore code QM2, in Installation 2, su Machine 1:
  - Canale mittente QM2.to.QM1
  - Canale ricevente QM1.to.QM2
  - Coda di trasmissione QM1
  - Un listener configurato sulla porta 1415
- Sul gestore code QM3, in Installation 3, su Machine 2:
  - Canale mittente QM3.to.QM1
  - Canale ricevente QM1.to.QM3
  - Coda di trasmissione QM1
  - Il listener predefinito

Una coda, Qon2 , è definita su QM2e una coda Qon3 è definita su QM3.

L'utente mquser è definito su entrambe le macchine, gli è stato concesso il ruolo MQWebAdmin in REST APIe gli è stata concessa l'autorizzazione ad accedere alle code appropriate su ciascun gestore code.

Il comando setmqweb viene utilizzato per configurare il gestore code QM1 come gestore code del gateway predefinito.

Il seguente diagramma mostra questa configurazione:

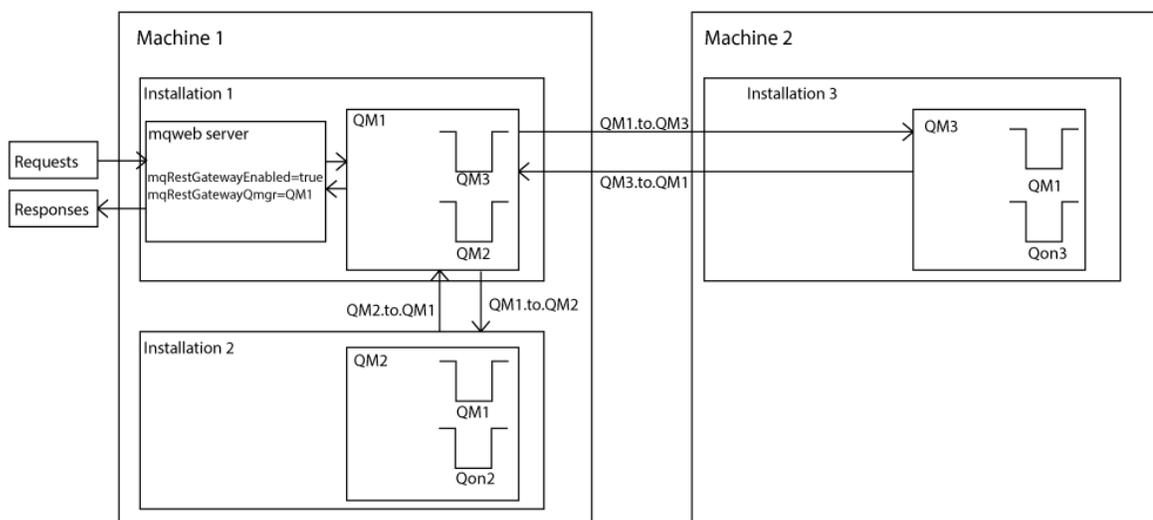


Figura 14. Diagramma di configurazione di esempio per la gestione remota utilizzando REST API.

La seguente richiesta REST viene inviata al server mqweb:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM2/queue?
attributes=general.isTransmissionQueue
```

Viene ricevuta la seguente risposta:

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false
    },
    "name": "Qon2",
    "type": "local"
  }
]
```

La seguente richiesta REST viene inviata al server mqweb:

```
GET https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM3/queue?
attributes=general.isTransmissionQueue,general.description
```

Viene ricevuta la seguente risposta:

```
{
  "queue" :
  [ {
    "general": {
      "isTransmissionQueue": true,
      "description": "Transmission queue for remote admin."
    },
    "name": "QM1",
    "type": "local"
  },
  {
    "general": {
      "isTransmissionQueue": false,
      "description": "A queue on QM3."
    },
    "name": "Qon3",
    "type": "local"
  }
]
```

## REST API data/ora

Quando le informazioni di data e ora vengono restituite da amministrative REST API, vengono restituite in UTC (Coordinated Universal Time) e in un formato impostato.

La data e l'ora vengono restituite nel seguente formato data/ora:

```
YYYY-MM-DDTHH:mm:ss:sssZ
```

Ad esempio, 2012-04-23T18:25:43.000Z, dove Z indica che il fuso orario è UTC (Coordinated Universal Time).

L'accuratezza di questa data / ora non è garantita. Ad esempio, se il server mqweb non viene avviato nello stesso fuso orario del gestore code specificato nell'URL della risorsa, la data / ora potrebbe non essere accurata. Inoltre, se sono necessarie regolazioni dell'ora legale, la data / ora potrebbe non essere accurata.

## REST API gestione degli errori

Il REST API riporta gli errori restituendo un codice di risposta HTTP appropriato, ad esempio 404 (Non trovato) e una risposta JSON. Qualsiasi codice di risposta HTTP non compreso nell'intervallo 200 - 299 viene considerato un errore.

### Il formato della risposta di errore

La risposta è in formato JSON nella codifica UTF-8 . Contiene oggetti JSON nidificati:

- Un oggetto JSON esterno che contiene un singolo array JSON denominato `error`.
- Ogni elemento nell'array è un oggetto JSON che rappresenta le informazioni su un errore. Ogni oggetto JSON contiene le proprietà riportate di seguito:

#### **tipo**

Stringa.

Il tipo di errore.

#### **messageId**

Stringa.

Un identificativo univoco per il messaggio nel formato MQWBnnnnX. Questo identificativo ha i seguenti elementi:

#### **MQWB**

Un prefisso che indica che il messaggio è stato originato nell'API REST IBM MQ .

#### **nnnn**

Un numero univoco che identifica il messaggio.

#### **X**

Una singola lettera che denota la severità del messaggio:

- I se un messaggio è puramente informativo.
- W se un messaggio indica un problema.
- E se un messaggio indica che si è verificato un errore.
- S se un messaggio indica che si è verificato un errore grave.

#### **Messaggio**

Stringa.

Una descrizione dell'errore.

#### **Spiegazione**

Stringa.

Una spiegazione dell'errore.

#### **azione**

Stringa.

Una descrizione dei passi che possono essere eseguiti per risolvere l'errore.

#### **qmgrName**

 Questo campo è disponibile solo per z/OS dove il gestore code è un membro del gruppo di condivisione code. È necessario che sia stato specificato il parametro di query facoltativo **commandScope** o l'attributo **queueSharingGroupDisposition** .

Stringa.

Il nome del gestore code che ha rilevato l'errore.

Questo campo non è applicabile per messaging REST API.

#### **completionCode**

Questo campo è disponibile solo quando **type** è `pcf`, `javao rest`.

Numero.

Il codice di completamento MQ associato all'errore.

#### **reasonCode**

Questo campo è disponibile solo quando **type** è pcf, javao rest.

Numero.

Il codice di errore MQ associato all'errore.

#### **eccezioni**

Questo campo è disponibile solo quando **type** è java.

Array.

Un array di eccezioni Java o JMS della catena. Ogni elemento dell'array di eccezioni contiene un array di stringhe **stackTrace**.

L'array di stringhe **stackTrace** contiene i dettagli di ciascuna eccezione suddivisa in linee.

**V 9.2.0**

Da IBM MQ 9.1.2, questo campo non viene più restituito.

## **Errori con i gruppi di condivisione code**

**z/OS**

In un gruppo di condivisione code, è possibile specificare un parametro di query facoltativo di **commandScope** per determinati comandi. Questo parametro consente la propagazione del comando ad altri gestori code nel gruppo di condivisione code. Uno di questi comandi può avere esito negativo in modo indipendente, determinando la riuscita di alcuni comandi e l'esito negativo di alcuni comandi per il gruppo di condivisione code.

Nei casi in cui un comando non riesce parzialmente, viene restituito un codice di errore HTTP di 500. Per ogni gestore code che ha generato un errore, le informazioni su tale errore vengono restituite come elemento nell'array JSON `error`. Per ogni gestore code che ha eseguito correttamente il comando, il nome del gestore code viene restituito come elemento in un array JSON `success`.

## **Esempi**

- Il seguente esempio mostra la risposta di errore a un tentativo di ottenere informazioni su un gestore code che non esiste:

```
"error": [
  {
    "type": "rest",
    "messageId": "MQWB0009E",
    "message": "MQWB0009E: Could not query the queue manager 'QM1'",
    "explanation": "The MQ REST API was invoked specifying a queue manager name which
cannot be located.",
    "action": "Resubmit the request with a valid queue manager name or no queue manager
name, to retrieve a list of queue managers. "
  }
]
```

**z/OS**

- Il seguente esempio mostra la risposta di errore ad un tentativo di eliminare una coda in un gruppo di condivisione code che non esiste per alcuni gestori code:

```
"error" : [
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason
code is 3312 : 'MQRCCF_UNKNOWN_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be
located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name
to retrieve a list of queues.",
    "qmgrName": "QM1"
  },
  {
    "type": "rest",
```

```

    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason code is 3312 : 'MQRCCF_UNKNOwn_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name to retrieve a list of queues.",
    "qmgrName": "QM2"
  }
],
"success" : [{"qmgrName": "QM3"}, {"qmgrName": "QM4"}]

```

## Errori con richieste MFT

Se i servizi MFT REST API non sono abilitati e si richiama MFT REST API, si riceve la seguente eccezione:

```

{"error": [{
  "action": "Enable the Managed File Transfer REST API and resubmit the request.",
  "completionCode": 0,
  "explanation": "Managed File Transfer REST calls are not permitted as the service is disabled.",
  "message": "MQWB0400E: Managed File Transfer REST API is not enabled.",
  "msgId": "MQWB0400E",
  "reasonCode": 0,
  "type": "rest"
}]}

```

Se i servizi MFT REST API sono abilitati e il gestore code di coordinamento non è impostato nel file `mqwebuser.xml`, si riceve la seguente eccezione:

```

{"error": [{
  "action": "Set the coordination queue manager name and restart the mqweb server.",
  "completionCode": 0,
  "explanation": "Coordination queue manager name must be set before using Managed File Transfer REST services.",
  "message": "MQWB0402E: Coordination queue manager name is not set.",
  "msgId": "MQWB0402E",
  "reasonCode": 0,
  "type": "rest"
}]}

```

## Rilevamento REST API

La Documentation per REST API è disponibile in IBM Documentation e in formato Swagger. Swagger è un approccio comunemente utilizzato per documentare le API REST. La documentazione Swagger per REST API può essere visualizzata abilitando la funzione di rilevamento API sul server mqweb.

### Prima di iniziare

**Importante:** La funzione di rilevamento API è stata stabilizzata ed è ancora possibile utilizzare questa funzione. Attualmente, IBM MQ non supporta l'utilizzo della funzione `mpOpenAPI`.

È necessario abilitare la sicurezza per il server mqWeb per visualizzare la documentazione Swagger utilizzando il rilevamento API. Per ulteriori informazioni sulla procedura richiesta per abilitare la sicurezza, consultare [IBM MQ Console e REST API sicurezza](#).

### Procedura

1. Individuare il file `mqwebuser.xml` in una delle seguenti directory:

-  `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
-  `WLP_user_directory/servers/mqweb`

Dove `WLP_user_directory` è la directory specificata quando lo script `crtmqweb` è stato eseguito per creare la definizione del server mqweb.

## 2. Aggiungere l'XML appropriato al file mqwebuser.xml :

- Se le tag <featureManager> sono presenti nel file mqwebuser.xml , aggiungere il seguente XML all'interno delle tag <featureManager> :  

```
<feature>apiDiscovery-1.0</feature>
```
- Se le tag <featureManager> non esistono nel proprio file mqwebuser.xml , aggiungere il seguente XML nelle tag <server> :

```
<featureManager>  
  <feature>apiDiscovery-1.0</feature>  
</featureManager>
```

## 3. Visualizzare la documentazione Swagger utilizzando uno dei seguenti metodi:

- Visualizzare una pagina Web che è possibile sfogliare e provare REST API immettendo il seguente URL in un browser:

`https://host:port/ibm/api/explorer`

Oltre all'autenticazione di ogni richiesta, è necessario includere un'intestazione `ibm-mq-rest-csrf-token` per ogni richiesta POST, PATCH o DELETE. Il contenuto di questa intestazione può essere qualsiasi stringa, incluso uno spazio.

Questa intestazione della richiesta viene utilizzata per confermare che le credenziali utilizzate per autenticare la richiesta vengono utilizzate dal proprietario delle credenziali. Ovvero, il token viene utilizzato per evitare attacchi cross - site request forgery.

- Richiama un singolo documento Swagger 2 che descrive l'intero REST API immettendo un HTTP GET al seguente URL:

`https://host:port/ibm/api/docs`

Questo documento può essere utilizzato per le applicazioni in cui si desidera navigare in modo programmatico nelle API disponibili.

### **host**

Specifica il nome host o indirizzo IP su cui è disponibile REST API .

Il valore predefinito è localhost.

### **porta**

Specifica il numero di porta HTTPS utilizzato da amministrative REST API .

Il valore predefinito è 9443.

Se il nome host o il numero di porta viene modificato rispetto al valore predefinito, è possibile determinare i valori corretti dall'URL REST API . Utilizzare il comando **dspmweb status** per visualizzare l'URL.

## **REST API supporto lingua nazionale**

REST API supporta, con determinate qualifiche, la possibilità di specificare le lingue nazionali come parte di una richiesta HTTP.

### **Sfondo**

Le intestazioni HTTP consentono di specificare un determinato comportamento nelle richieste e di fornire ulteriori informazioni nelle risposte.

Nelle intestazioni HTTP è inclusa la possibilità di richiedere che le informazioni vengano restituite in una lingua nazionale. Il REST API rispetta questa intestazione dove possibile.

## Specifica di una lingua nazionale

Nell'intestazione HTTP ACCEPT - LANGUAGE, è possibile fornire una o più tag di lingua. È possibile, facoltativamente, associare una classificazione ai tag, consentendo la specifica di un elenco ordinato per preferenza. [Questa pagina](#) contiene un'utile discussione sul principio.

REST API rispetta questa intestazione, selezionando una lingua dall'intestazione ACCEPT - LANGUAGE e restituendo i messaggi in tale lingua. Quando l'intestazione ACCEPT - LANGUAGE non contiene alcuna lingua supportata da REST API, i messaggi vengono restituiti in una lingua predefinita. Questa lingua predefinita corrisponde alla locale predefinita del server Web REST API.

La sezione [“Quali dati vengono tradotti?”](#) a pagina 87 spiega quali dati vengono tradotti.

## Indicazione della lingua applicabile nelle risposte

L'intestazione HTTP CONTENT - LANGUAGE nelle risposte da REST API indica la lingua in cui vengono restituiti i messaggi.

## Quali dati vengono tradotti?

I messaggi di errore e informativi sono tradotti, mentre l'altro testo non lo è.

- I dati restituiti da un gestore code non vengono tradotti, ad esempio nel caso di esecuzione di un comando MQSC tramite REST API, le risposte del gestore code sono nella locale del gestore code.
- La documentazione generata (Swagger) per REST API, come esposta tramite la funzione `apiDiscovery`, è in inglese.

## Quali lingue sono supportate?

Oltre all'inglese, i messaggi di errore e informativi di REST API sono tradotti nelle seguenti lingue.

### **Cinese (semplificato)**

Contrassegnato dalla tag della lingua `zh_CN`

### **Cinese (tradizionale)**

Contrassegnato dalla tag della lingua `zh_TW`

### **Ceco**

Contrassegnato dalla tag della lingua `cs`

### **Franco francese**

Contrassegnato dalla tag della lingua `fr`

### **Ungherese**

Contrassegnato dalla tag della lingua `hu`

### **:NONE.**

Contrassegnato dalla tag della lingua `it`

### **Giapponese**

Contrassegnato dalla tag della lingua `ja`

### **Coreano**

Contrassegnato dalla tag della lingua `ko`

### **Polacco**

Contrassegnato dalla tag della lingua `pl`

### **(Brasiliano) Portoghese**

Contrassegnato dalla tag della lingua `pt_BR`

### **Russo**

Contrassegnato dalla tag della lingua `ru`

### **Spagnolo**

Contrassegnato dalla tag della lingua `es`

## Esempi

Negli esempi, il server Web ha una locale predefinita in inglese.

### Specifica di una singola lingua supportata

Nelle intestazioni della richiesta, ACCEPT-LANGUAGE è impostata su `fr`. Questa impostazione specifica che il francese è la lingua preferita per il testo traducibile.

Nelle intestazioni della risposta, CONTENT-LANGUAGE è impostata su `fr`. Questa impostazione indica che i messaggi di errore e informativi nella risposta sono in francese.

### Specifica di un elenco di lingue

Nelle intestazioni della richiesta, ACCEPT-LANGUAGE è impostata su `am, fr`. Questa impostazione specifica che l'amarico e il francese sono lingue accettabili per il testo traducibile e che l'amarico è la lingua preferita per il testo traducibile.

Nelle intestazioni della risposta, CONTENT-LANGUAGE è impostata su `fr`. Questa impostazione indica che i messaggi di errore e informativi nella risposta sono in francese, poiché REST API non supporta l'amarico.

### Specifica di una singola lingua non supportata

Nelle intestazioni della richiesta, ACCEPT-LANGUAGE è impostata su `am`. Questa impostazione specifica che l'amarico è la lingua preferita per il testo traducibile.

Nelle intestazioni della risposta, CONTENT-LANGUAGE è impostata su `en`. Questa impostazione indica che i messaggi di errore e informativi nella risposta sono in inglese, poiché REST API non supporta l'amarico.

## REST API versioni

Il numero di versione REST API fa parte dell'URL di base per le richieste REST. Ad esempio, `https://localhost:9443/ibmmq/rest/v2/admin/installation`. Il numero di versione viene utilizzato per isolare i client dalle modifiche al REST API che potrebbero essere introdotte nelle release future.

**V 9.2.0** IBM MQ 9.2.0 introduce la versione 2 di REST API. Questo aumento di versione si applica a administrative REST API, messaging REST API e MFT REST API. Questo aumento di versione modifica l'URL della risorsa utilizzato per REST API. Il prefisso URL per gli URL della risorsa alla Versione 2 è il seguente URL:

```
https://host:port/ibmmq/rest/v2/
```

Alcune modifiche introdotte in REST API potrebbero modificare la funzione REST API esistente in modo che i client che utilizzano REST API potrebbero dover essere aggiornati. Per evitare che tali modifiche forzino l'aggiornamento dei client, il numero di versione di REST API viene aumentato e la funzione esistente viene stabilizzata al numero precedente. La nuova funzione che potrebbe modificare la funzione esistente viene aggiunta a REST API al nuovo numero di versione. Di conseguenza, i client possono continuare a utilizzare REST API alla versione precedente senza essere aggiornati.

Le modifiche REST API che potrebbero comportare la richiesta di un aggiornamento client includono le seguenti modifiche:

- Rimozione del supporto per un attributo esistente nel JSON inviato o restituito da REST API.
- Rimozione di un URL, un verbo HTTP o un'intestazione. Ad esempio, se viene rinominato un URL o un'intestazione o se viene utilizzato un verbo diverso.
- Aggiunta di un nuovo attributo JSON obbligatorio ai dati inviati a un URL esistente.
- Aggiunta di una nuova intestazione HTTP obbligatoria ai dati inviati a un URL esistente.
- Aggiunta di un nuovo parametro di query obbligatorio a un URL esistente.

Quando questo tipo di modifica viene introdotto nella funzione REST API che esisteva in una release Long Term Support (LTS), il numero di versione di REST API viene aumentato per la prima di queste modifiche. Eventuali modifiche successive apportate all'interno di una release di Continuous Delivery (CD)

che potrebbero richiedere modifiche ai client che utilizzano REST API utilizzano il numero di versione nuovo.

Questo numero di versione rimane lo stesso nelle release successive di CD fino alla release successiva di LTS . Pertanto, il numero di versione aumenta al massimo una volta tra le release di LTS .

Quando il numero di versione viene aumentato, la funzione REST API esistente viene stabilizzata al vecchio numero di versione. Vale a dire, la funzione REST API esistente che era disponibile nella release LTS rimane disponibile al vecchio numero di versione, ma non vengono apportate ulteriori modifiche a quella versione. Qualsiasi nuova funzione aggiunta a REST API viene aggiunta alla nuova versione di REST API . Tuttavia, eventuali aggiunte apportate a REST API nelle release CD prima dell'aumento della versione non sono garantite per essere incluse nella versione precedente di REST API.

I client esistenti possono continuare ad utilizzare REST API al vecchio numero di versione senza richiedere alcuna modifica. Le versioni precedenti di REST API potrebbero essere obsolete ed eventualmente rimosse.

Alcune modifiche non richiedono modifiche ai client che utilizzano REST API. Queste modifiche non comportano un aumento del numero di versione. Pertanto, assicurarsi che qualsiasi client che utilizza REST API non debba essere aggiornato quando vengono introdotti questi tipi di modifiche. Queste modifiche al file REST API potrebbero includere le seguenti modifiche:

- Aggiunta di un nuovo attributo JSON ai dati esistenti restituiti da REST API.
- Aggiunta di un nuovo URL.
- Aggiunta di un nuovo verbo HTTP a un URL esistente.
- Aggiunta di un nuovo codice di stato a un URL esistente.
- Aggiunta di nuovi attributi JSON facoltativi ai dati inviati a un URL esistente.
- Aggiunta di nuovi parametri di query su un URL esistente.
- Aggiunta di nuove intestazioni ai dati inviati a un URL esistente.
- Ritorno di nuove intestazioni da REST API.

## **Modifiche alla nuova funzione API REST Continuous Delivery**

Per la nuova funzione REST API aggiunta in una release CD , tutte le modifiche apportate a questa nuova funzione che potrebbero richiedere modifiche ai client REST API non aumentano il numero di versione. In altre parole, la nuova funzione può essere modificata prima della release successiva di LTS senza aumentare il numero di versione. Quando la funzione è inclusa in una release LTS , qualsiasi modifica successiva che potrebbe richiedere modifiche ai client REST API aumenta il numero di versione.

### **Esempio**

1. In LTS release X, REST API è alla versione 1.
2. In CD release X.0.1, viene aggiunto il supporto per un nuovo URL. Questa modifica non richiede modifiche ai client che utilizzano REST API. Pertanto, REST API rimane alla versione 1.
3. In CD X.0.2, viene aggiunto il supporto per un nuovo URL. Questa modifica non richiede modifiche ai client che utilizzano l'API REST. Pertanto, REST API rimane alla versione 1.
4. Alla release LTS Y, REST API è alla versione 1.
5. In CD release Y.0.1, viene ridenominato un URL esistente. Questa modifica potrebbe richiedere modifiche ai client che utilizzano REST API. Pertanto, una nuova versione di REST API viene creata come versione 2. L'URL ridenominato è incluso nella versione 2 di REST API, insieme a tutta la funzione esistente. Tutte le nuove funzioni aggiunte a REST API vengono aggiunte alla versione 2. La versione 1 rimane stabilizzata al livello di LTS release Y.
6. In CD release Y.0.2, un altro URL esistente viene ridenominato. Poiché la versione è già aumentata in CD release Y, REST API rimane alla versione 2. La versione 1 rimane stabilizzata al livello di LTS release Y.

7. Nella release LTS Z, REST API rimane alla versione 2. La versione 1 rimane stabilizzata al livello di LTS release Y.

## Gestione mediante una console Web

È possibile eseguire attività di gestione di base utilizzando una console web.

**V 9.2.0** Da IBM MQ 9.2.0, è disponibile una nuova console Web, nota come New Web Console, consultare [“Presentazione rapida della nuova console Web”](#) a pagina 92.

Se si desidera, è possibile continuare a utilizzare Dashboard Web Console su Multiplatforms, consultare [“Passaggio da un tipo di console all'altro”](#) a pagina 112.

**Nota:** Non disabilitare il server dei comandi su nessuno dei gestori code quando si utilizza una console Web. Se il server dei comandi è disabilitato per un gestore code, la console Web non risponde con lunghi ritardi all'elaborazione dei comandi. I comandi emessi per il gestore code per cui il server dei comandi è disabilitato vanno in timeout.

### Attività correlate

**V 9.2.0** [Traccia della nuova console Web](#)  
[Traccia della console Web del dashboard](#)

## **V 9.2.0** Introduzione alla console Web

Introduzione rapida alla console Web.

### Prima di iniziare

Per completare questa attività, è necessario essere un utente con particolari privilegi in modo da poter utilizzare il comando **dspmqweb**:

- **z/OS** Su z/OS, è necessario disporre dell'autorità per eseguire il comando **dspmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
- **Multi** Su tutti gli altri sistemi operativi, è necessario essere un utente privilegiato.

**IBM i** Su IBM i, i comandi devono essere in esecuzione in QSHELL.

### Procedura

1. Se il server mqWeb non è ancora configurato per l'utilizzo da parte della console Web, configurare il server mqWeb.

Per ulteriori informazioni sulla creazione di una configurazione di base per il server mqweb con un registro di base, consultare [Configurazione di base per il server mqweb](#).

2. **z/OS** Su z/OS, impostare la variabile di ambiente `WLP_USER_DIR` in modo da poter utilizzare il comando **dspmqweb**. Impostare la variabile in modo che punti alla configurazione del server mqweb immettendo il seguente comando:

```
export WLP_USER_DIR=WLP_user_directory
```

dove `WLP_user_directory` è il nome della directory passata a `crtmqweb`. Ad esempio:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Per ulteriori informazioni, consultare [Creazione del server mqweb](#).

3. Determinare l'URI per la console Web immettendo il seguente comando:

```
dspmqweb status
```

Il comando genera un output simile al seguente:

```
MQWB1124I: Server 'mqweb' is running.  
URLS:  
https://localhost:9443/ibmmq/rest/v1/  
https://localhost:9443/ibmmq/console/
```

L'URI della console Web termina con il suffisso `console/`.

4. Connettersi alla console Web immettendo l'URL da “3” a pagina 90 in un browser.

Il browser potrebbe produrre un'eccezione di sicurezza perché il certificato predefinito fornito con il server mqweb non è un certificato attendibile. Scegliere di passare alla console Web.

5. Accedere alla console Web. Utilizzare il nome utente `mqadmin` e la password `mqadmin`.

## Operazioni successive

Per default, la console web utilizza l'autenticazione basata su token per autenticare gli utenti. È anche possibile utilizzare l'autenticazione certificato client. Per ulteriori informazioni, consultare [Utilizzo dell'autenticazione del certificato client con REST API e la console web](#).

## Restrizioni su z/OS

Le seguenti limitazioni si applicano quando si utilizza IBM MQ Console per gestire i gestori code su z/OS.

- I gestori code su z/OS non possono essere creati, eliminati, avviati o arrestati.
- Gli iniziatori di canali su z/OS non possono essere avviati o arrestati e lo stato dell'iniziatore di canali non viene visualizzato.
- I listener non possono essere visualizzati o gestiti.
- I comandi start, ping, resolve e reset channel possono essere emessi solo con CHLDISP (DEFAULT).
- I nuovi oggetti possono essere creati solo con QSGDISP (QMGR).
- Gli oggetti definiti con QSGDISP (GROUP) non possono essere visualizzati o gestiti.
- Impossibile gestire la sicurezza del gestore code.
- L'utilizzo delle risorse di sistema non può essere monitorato.

### Concetti correlati

[“Gestione mediante una console Web” a pagina 90](#)

È possibile eseguire attività di gestione di base utilizzando una console web.

### Attività correlate

[“Utilizzo dei gestori code locali” a pagina 95](#)

I gestori code locali vengono creati, configurati e controllati dal livello superiore della vista Gestisci



## Restrizioni su IBM MQ for Multiplatforms

Le seguenti limitazioni si applicano quando si utilizza IBM MQ Console per gestire i gestori code su IBM MQ for Multiplatforms.

- Non è possibile utilizzare IBM MQ Console per utilizzare i canali AMQP.
- Non è possibile utilizzare IBM MQ Console per utilizzare i canali MQTT.

### Concetti correlati

[“Gestione mediante una console Web” a pagina 90](#)

È possibile eseguire attività di gestione di base utilizzando una console web.

## Attività correlate

[“Utilizzo dei gestori code locali” a pagina 95](#)

I gestori code locali vengono creati, configurati e controllati dal livello superiore della vista Gestisci

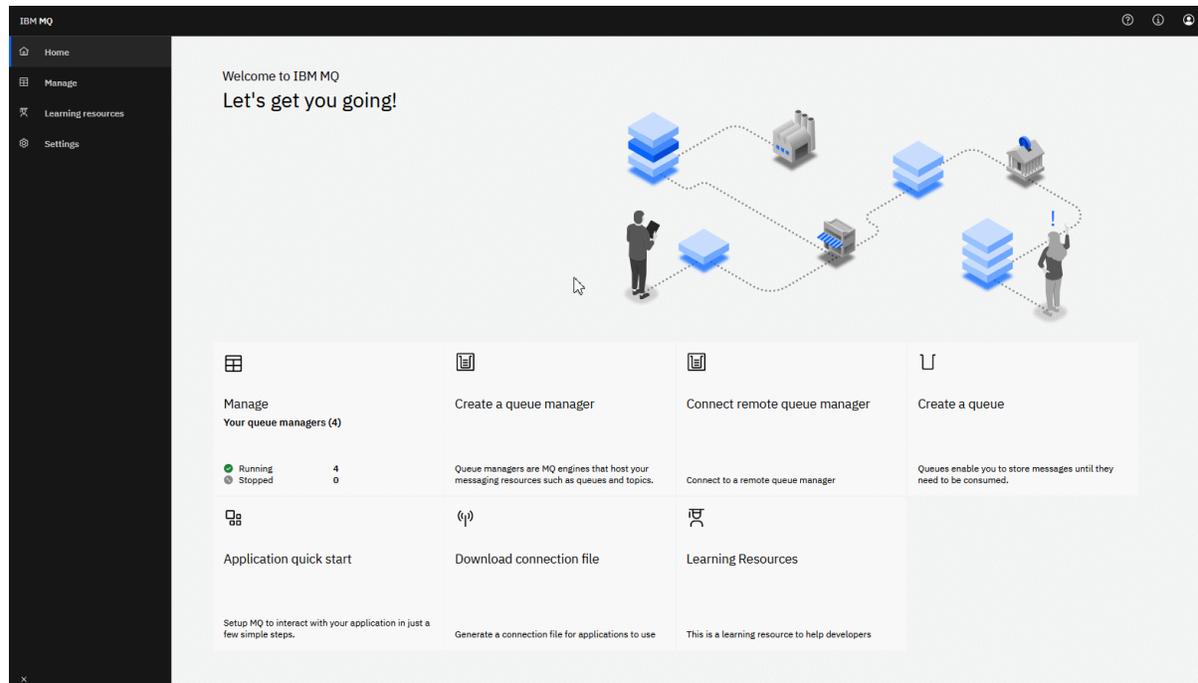


## V 9.2.0 Presentazione rapida della nuova console Web

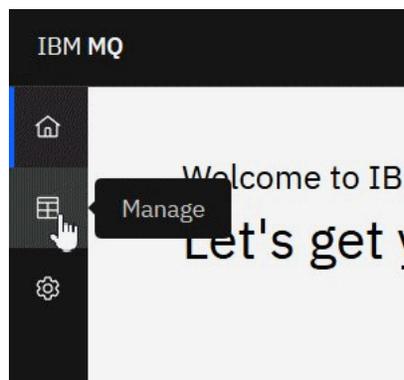
La presentazione rapida consente di iniziare con New Web Console. Seguono argomenti più dettagliati sul suo utilizzo.

New Web Console è l'IU web preferita, ma se vuoi continuare a utilizzare la console esistente ( Dashboard Web Console), puoi tornare a tale interfaccia, vedi [“Passaggio da un tipo di console all'altro” a pagina 112.](#)

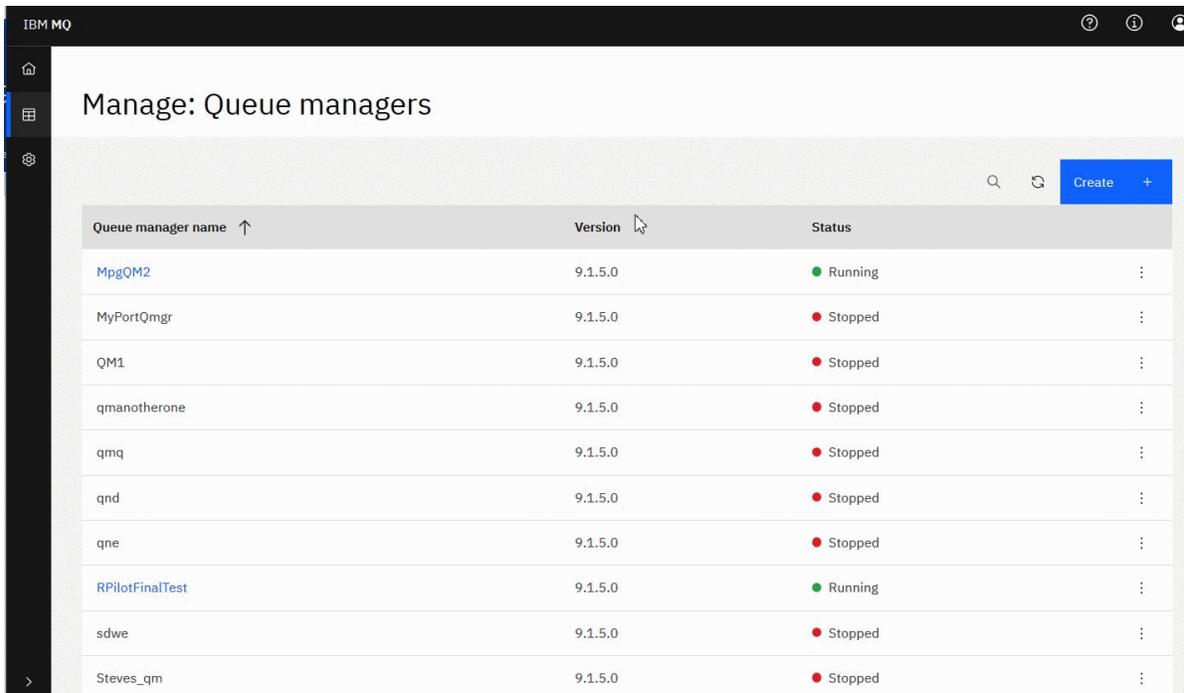
Quando si accede per la prima volta a New Web Console , si viene portati alla pagina di destinazione. Da qui è possibile scegliere di gestire i gestori code esistenti, creare un gestore code o una coda, passare ad alcuni argomenti di formazione o aprire le informazioni sul prodotto IBM MQ in IBM Documentation. È anche possibile avviare l'avvio rapido dell'applicazione, che guida l'utente attraverso il processo di impostazione rapida e semplice della messaggistica tra gestori code e applicazioni nuovi o esistenti.



In alternativa, è possibile semplicemente fare clic sull'icona Gestisci per avviare la gestione degli oggetti IBM MQ .

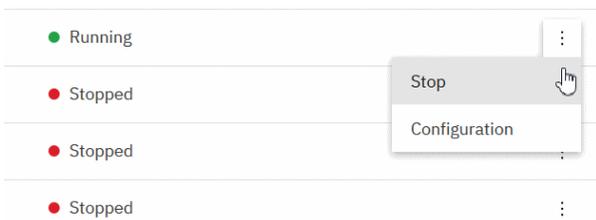


La vista di gestione mostra inizialmente i gestori code e il relativo stato corrente. È inoltre possibile creare nuovi gestori code e connettere i gestori code remoti.



Queue manager name ↑	Version	Status
MpgQM2	9.1.5.0	Running
MyPortQmgr	9.1.5.0	Stopped
QM1	9.1.5.0	Stopped
qmanotherone	9.1.5.0	Stopped
qmq	9.1.5.0	Stopped
qnd	9.1.5.0	Stopped
qne	9.1.5.0	Stopped
RPilotFinalTest	9.1.5.0	Running
sdwe	9.1.5.0	Stopped
Steves_qm	9.1.5.0	Stopped

Ogni gestore code dispone di un menu che consente di arrestare o configurare un gestore code in esecuzione, oppure di avviare o eliminare un gestore code arrestato.



I record di autorizzazione, gli oggetti Informazioni di autenticazione e i record di autenticazione di canale per il gestore code si trovano nella scheda **Sicurezza** della pagina **Configurazione** del gestore code, in cui è possibile crearne e aggiungerne di nuovi.

Fare clic sul nome di un gestore code in esecuzione per aprirne il dashboard.

IBM MQ

Manage / MpgQM

## Queue manager: MpgQM

Configuration

Queues Topics Subscriptions Communication

1 Full capacity  
1 user-defined  
0 system

2 Greater than 50% capacity  
1 user-defined  
1 system

61 Less than 1% capacity  
15 user-defined  
46 system

Search Filter Refresh Create +

Name ↑	Type	Depth %	Maximum depth
AMQ.MQEXPLORER.5E68C78B20805304	Local	0%	0/5000
Queueue1	Local	0%	0/1
Queueue2	Local	0%	0/5000
Queueue3	Local	0%	0/5000
Queueue4	Local	0%	0/5000

Dal dashboard del gestore code è possibile completare le seguenti azioni:

Nella scheda **Code** :

- Crea nuove code
- Configura code esistenti: 
- Fare clic sul nome di una coda per visualizzare i messaggi esistenti e crearne di nuovi

Nella scheda **Argomenti** :

- Crea nuovi argomenti
- Configura argomenti esistenti 
- Fare clic su un nome argomento per visualizzare le sottoscrizioni corrispondenti

Nella scheda **Sottoscrizioni** :

- Crea nuove sottoscrizioni gestite o non gestite
- Configura sottoscrizioni esistenti 

Nella scheda **Comunicazioni** :

- Listener:
  - Avviare, arrestare e configurare i listener 
  - Crea nuovi listener
- Canali gestore code:
  - Avvio, arresto, ping e configurazione dei canali 
  - Creare nuovi canali
  - Reimposta canali (dalla voce di menu **Avanzate**)
  - Risolvere i messaggi in dubbio sui canali (dalla voce di menu **Avanzate**)

- Canali applicazione:
  - Avvio, arresto, ping e configurazione dei canali 
  - Creare nuovi canali
  - Reimposta canali (dalla voce di menu **Avanzate**)
  - Risolvere i messaggi in dubbio sui canali (dalla voce di menu **Avanzate**)

## Utilizzo dei gestori code locali

I gestori code locali vengono creati, configurati e controllati dal livello superiore della vista Gestisci



### Informazioni su questa attività

 **Multi** La vista Gestisci elenca i gestori code locali aggiunti all'installazione di IBM MQ da cui è in esecuzione IBM MQ Console . I gestori code associati a diverse installazioni di IBM MQ nello stesso sistema non sono elencati.

 **z/OS** Su z/OS, la vista Gestisci elenca i gestori code che sono alla stessa versione di IBM MQ Consolee sono definiti sul sistema su cui è in esecuzione MQ Console . I gestori code con una versione diversa rispetto a MQ Console non sono elencati.

È possibile selezionare singoli gestori code dall'elenco da utilizzare.

**Nota:** IBM MQ Console non supporta i gestori code di dati replicati (RDQM).

### Procedura

- Per creare un nuovo gestore code locale:
  - Fare clic sul pulsante **Crea**  nella vista di elenco del gestore code.
  - Immettere un nome per il nuovo gestore code. Il nome può contenere un massimo di 48 caratteri. I caratteri validi sono lettere e numeri e i caratteri ":", "/", "\_e "%".
  - Opzionale: Immettere una porta TCP/IP disponibile per il gestore code su cui ascoltare. Il numero di porta non deve essere superiore a 65535.
  - Fai clic su **Crea**. Il nuovo gestore code viene creato e avviato.
- Per avviare un gestore code locale:
  - Individuare il gestore code che si desidera avviare nell'elenco.
  - Selezionare **Avvia** dal menu .
- Per arrestare un gestore code locale:
  - Selezionare il gestore code che si desidera arrestare dall'elenco nel widget gestore code locale.
  - Selezionare **Arresta** dal menu .
- Per cancellare un gestore code locale:
  - Se il gestore code è in esecuzione, arrestarlo.
  - Seleziona **Configure** dal menu  e seleziona **Delete queue manager**.

- c) Confermare che si desidera eliminare il gestore code immettendone il nome nella finestra di conferma. Il gestore code e tutti gli oggetti associati vengono eliminati.
- Per visualizzare e modificare le proprietà di un gestore code locale:
  - a) Verificare che il gestore code sia in esecuzione e individuarlo nell'elenco dei gestori code.
  - b) Selezionare **Configura** dal menu  .
  - c) Verificare che la pagina **Proprietà** sia selezionata. Visualizzare le proprietà e modificarle come richiesto. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi. Per informazioni su una proprietà, è possibile visualizzare le informazioni sulla proprietà in [Proprietà del gestore code](#).
- Per lavorare con le impostazioni di protezione per il gestore code locale:
  - a) Verificare che il gestore code sia in esecuzione e selezionarlo dall'elenco dei gestori code.
  - b) Selezionare **Configura** dal menu  .
  - c) Assicurarsi che la scheda **Sicurezza** sia selezionata.
  - d) È possibile gestire oggetti di autenticazione, record di autorizzazione o oggetti di autenticazione di canale. Per ulteriori informazioni, consultare i seguenti argomenti:
    - [“Utilizzo degli oggetti delle informazioni di autenticazione”](#) a pagina 96
    - [“Gestione dei record di autorizzazione del gestore code”](#) a pagina 98
    - [“Utilizzo dei record di autenticazione di canale”](#) a pagina 99

### **V 9.2.0** *Utilizzo degli oggetti delle informazioni di autenticazione*

È possibile utilizzare la console per aggiungere ed eliminare gli oggetti delle informazioni di autenticazione su un gestore code. È inoltre possibile visualizzare e impostare le proprietà e gestire i record di autorizzazione per gli oggetti.

### **Informazioni su questa attività**

La vista delle informazioni di autenticazione elenca le informazioni di autenticazione esistenti per uno specifico gestore code. È possibile selezionare singole informazioni di autenticazione dall'elenco da gestire.

Le informazioni di autenticazione del gestore code fanno parte del supporto IBM MQ per TLS (Transport Layer Security). Questi oggetti contengono le definizioni richieste per eseguire il controllo della revoca dei certificati utilizzando OCSP o CRL (Certificate Revocation Lists) sui server LDAP e le definizioni richieste per abilitare il controllo dell'ID utente e della password.

### **Procedura**

- Per visualizzare le informazioni di autenticazione per un gestore code:
  - a) Verificare che il gestore code sia in esecuzione e selezionarlo dall'elenco dei gestori code.
  - b) Selezionare **Configura** dal menu  .
  - c) Assicurarsi che la scheda **Sicurezza** sia selezionata.
  - d) Selezionare **Informazioni di autenticazione** dal pannello di navigazione.
- Per aggiungere un oggetto delle informazioni di autenticazione:
  - a) Fare clic sul pulsante Crea  nella vista elenco delle informazioni di autenticazione.
  - b) Specificare il nome dell'oggetto relativo alle informazioni di autenticazione. I caratteri validi sono lettere e numeri e i caratteri ".", "/", "\_", "e" "%".

- c) Specificare il tipo di oggetto delle informazioni di autenticazione.
- d) Specificare ulteriori informazioni appropriate per il tipo di oggetto:
- Per **CRL LDAP**, specificare il **nome server LDAP**. Questo nome è il nome host, l'indirizzo decimale con punti IPv4 o la notazione esadecimale IPv6 dell'host su cui è in esecuzione il server LDAP, con un numero di porta facoltativo. Facoltativamente, è possibile specificare un nome utente e una password per l'utente che accede al server LDAP.
  - Per **OCSP**, specificare l' **URL responder OCSP**. Questo URL è l'URL del responder utilizzato per controllare la revoca del certificato. Questo valore deve essere un URL HTTP contenente il nome host e il numero di porta del responder OCSP. Se il responder OCSP utilizza la porta 80, che è il valore predefinito per HTTP, il numero di porta può essere omissso. Gli URL HTTP sono definiti in RFC 1738.
  - Per **IDPW OS**, non ci sono ulteriori requisiti anche se è possibile specificare facoltativamente ulteriori opzioni per questo tipo di autenticazione.
  - Per **IDPW LDAP**, specificare il **Nome server LDAP** e il nome **Utente breve** . Il nome del server LDAP è il nome host, l'indirizzo decimale puntato IPv4 o la notazione esadecimale IPv6 dell'host su cui è in esecuzione il server LDAP, con un numero di porta facoltativo. Il nome utente breve è il campo nel record utente LDAP utilizzato come nome breve per la connessione. Facoltativamente, è possibile specificare ulteriori opzioni per questo tipo di autenticazione.
- e) Fare clic su **Aggiungi**.
- Per eliminare un oggetto delle informazioni di autenticazione:
    - a) Selezionare l'icona di spanner  per l'oggetto delle informazioni di autenticazione che si desidera eliminare dall'elenco.
    - b) Nella vista proprietà dell'oggetto, fare clic su **Elimina oggetto informazioni di autenticazione**.
    - c) Confermare che si desidera eliminare l'oggetto delle informazioni di autenticazione facendo clic su **Elimina**. L'oggetto viene eliminato.
  - Per visualizzare e modificare le proprietà di un oggetto delle informazioni di autenticazione:
    - a) Selezionare l'icona della chiave  per l'oggetto delle informazioni di autenticazione che si desidera visualizzare dall'elenco.
    - b) Per modificare le proprietà visualizzate, fare clic sul pulsante Modifica .
    - c) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi.
    - d) Fare clic su **Salva** per salvare le modifiche.
  - Per visualizzare e modificare i record di autorizzazione per un oggetto delle informazioni di autenticazione:
    - a) Selezionare l'icona della chiave  per l'oggetto delle informazioni di autenticazione per cui si desidera visualizzare il record di autorità dall'elenco.
    - b) Selezionare la scheda **Sicurezza** .
    - c) Per modificare o eliminare un record di autorizzazione esistente, selezionare **Modifica** o **Elimina** nel menu  .
    - d) Per aggiungere un nuovo record di autorizzazione, fare clic sul pulsante **Aggiungi** , fornire i dettagli del nuovo record di autorizzazione e fare clic su **Crea**.

## Gestione dei record di autorizzazione del gestore code

È possibile controllare l'accesso degli utenti e dei gruppi ai gestori code specificando un record di autorizzazione per tale utente o gruppo.

### Informazioni su questa attività

È possibile ottimizzare l'accesso di un utente di messaggistica o di un gruppo di utenti di messaggistica a un determinato gestore code utilizzando i record di autorizzazione. Esistono due tipi di record di autorizzazione: i record di autorizzazione che controllano le autorità generali e i record di autorizzazione che controllano quali utenti e gruppi possono creare oggetti per il gestore code.

### Procedura

- Per visualizzare i record di autorizzazioni per un gestore code:
  - a) Verificare che il gestore code sia in esecuzione e selezionarlo dall'elenco dei gestori code.
  - b) Selezionare **Configura** dal menu .
  - c) Assicurarsi che la scheda **Sicurezza** sia selezionata.
  - d) Selezionare **Record di autorizzazioni** dal pannello di navigazione. La vista mostra i record di autorizzazioni in due riquadri, consentendo di gestire i record di autorizzazioni generali e di creare record di autorizzazioni.
- Per aggiungere un record di autorizzazione generale:
  - a) Fare clic sul pulsante Aggiungi  nella vista elenco dei record di autorizzazione.
  - b) Scegliere se si sta aggiungendo un record di autorizzazione per un utente o un gruppo.
  - c) Specificare il nome dell'utente o del gruppo per cui si sta aggiungendo un record di autorizzazione (il record di autorizzazione utilizza questo come nome)
  - d) Selezionare le autorizzazioni che si desidera concedere (consultare -- per ulteriori informazioni sulle autorizzazioni)
  - e) Fai clic su **Crea**.
- Per aggiungere un record di autorizzazione di creazione:
  - a) Fare clic sul pulsante Aggiungi  nella vista elenco Crea record di autorizzazione.
  - b) Scegliere se si sta aggiungendo un record di autorizzazione per un utente o un gruppo.
  - c) Specificare il nome dell'utente o del gruppo per cui si sta aggiungendo un record di autorizzazione (il record di autorizzazione utilizza questo come nome)
  - d) Selezionare i tipi di oggetto che si sta concedendo l'autorizzazione alla creazione.
  - e) Fai clic su **Crea**.
- Per eliminare un record di autorizzazioni:
  - a) Aprire il menu  per il record di autorizzazioni che si desidera eliminare e selezionare **Elimina**.
  - b) Confermare che si desidera eliminare l'oggetto delle informazioni di autenticazione facendo clic su **Elimina**. L'oggetto viene eliminato.
- Per visualizzare e modificare le proprietà di un record di autorizzazione:
  - a) Aprire il menu  per il record di autorizzazione che si desidera eliminare e selezionare **Modifica**.
  - b) Modificare le impostazioni come richiesto e fare clic su **Salva** per salvare le modifiche.

## V 9.2.0 Utilizzo dei record di autenticazione di canale

È possibile utilizzare IBM MQ Console per aggiungere ed eliminare record di autenticazione di canale su un gestore code. È inoltre possibile visualizzare e impostare le proprietà per i record di autenticazione di canale.

### Informazioni su questa attività

Per esercitare un controllo più preciso sull'accesso concesso ai sistemi di connessione a livello di canale, è possibile utilizzare i record di autenticazione di canale.

Per applicare la sicurezza, è possibile utilizzare i record di autenticazione di canale di blocco per bloccare l'accesso ai canali. È anche possibile utilizzare i record di autenticazione di canale della mappa di indirizzi per consentire l'accesso agli utenti specificati. Per ulteriori informazioni sui record di autenticazione di canale, consultare [Record di autenticazione di canale](#).

### Procedura

- Per visualizzare le informazioni di autenticazione di canale per un gestore code:
  - a) Verificare che il gestore code sia in esecuzione e selezionarlo dall'elenco dei gestori code.

- b) Selezionare **Configura** dal menu .
- c) Assicurarsi che la scheda **Sicurezza** sia selezionata.
- d) Selezionare **Autenticazione di canali** dal pannello di navigazione.

- Per aggiungere un record di autenticazione di canale:

- a) Fare clic sul pulsante Crea  nella vista elenco delle informazioni di autenticazione del canale.
- b) Scegliere il tipo di regola che si desidera utilizzare. Selezionare un **Consenti, Bloccao Avvisa**.
- c) Scegliere il tipo di identità per cui si sta configurando una regola di autenticazione di canale. Sono disponibili diversi tipi di identità, a seconda del tipo di regola selezionato.
- d) Fornire le informazioni richieste per l'identità che si sta specificando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.

- e) Fare clic sul pulsante Crea  per creare il record di autenticazione di canale.

Per ulteriori informazioni sulle impostazioni disponibili per i record di autenticazione di canale, consultare [Record di autenticazione di canale](#) e [SET CHLAUTH](#)

- Per eliminare un record di autenticazione di canale:

- a) Fare clic sull'icona della chiave  accanto al record di autenticazione di canali che si desidera eliminare.
- b) Nella vista Modifica autenticazione canale, fare clic su **Elimina oggetto autenticazione canale**.
- c) Confermare che si desidera eliminare il record di autenticazione di canale facendo clic su **Elimina**. Il record di autenticazione di canale viene eliminato.

- Per visualizzare e modificare le proprietà di un record di autenticazione di canale:

- a) Fare clic sull'icona della chiave  accanto al record di autenticazione di canale che si desidera modificare o visualizzare. Vengono visualizzate le proprietà.



- b) Fare clic sul pulsante Modifica
- c) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi.
- d) Fare clic su **Salva** per salvare le modifiche.

### V 9.2.3 Aggiunta di un gestore code remoto

È possibile utilizzare IBM MQ Console per gestire un gestore code in esecuzione su un sistema remoto.

#### Prima di iniziare

È necessario preparare il gestore code sul sistema remoto in modo che possa essere gestito in remoto, consultare il passo “1” a pagina 101 di [“Connessione di un gestore code remoto a MQ Console utilizzando la riga comandi” a pagina 101](#).

È inoltre necessario impostare un file di configurazione che controlli il modo in cui è possibile utilizzare connessioni remote da IBM MQ Console. Il file di configurazione viene creato utilizzando il comando **setmqweb** con il parametro **remote** (consultare [Configurazione del funzionamento della connessione del gestore code remoto e setmqweb](#)). Non è possibile modificare direttamente il file di configurazione.

#### Informazioni su questa attività

Utilizzare una CCDT (client connection definition table) in formato JSON per specificare i dettagli della connessione remota. Puoi creare una CCDT JSON utilizzando un editor di testo (vedi passo “2” a pagina 102 di [“Connessione di un gestore code remoto a MQ Console utilizzando la riga comandi” a pagina 101](#)) oppure puoi crearne una utilizzando IBM MQ Console.

In alternativa, è possibile creare la CCDT da IBM MQ Console specificando i dettagli di connessione direttamente quando si aggiunge il gestore code remoto.

È inoltre possibile connettere un gestore code remoto a IBM MQ Console utilizzando la riga di comando per tutte le attività richieste (in aggiunta alla preparazione del gestore code remoto e alla creazione di un CCDT). Consultare [“Connessione di un gestore code remoto a MQ Console utilizzando la riga comandi” a pagina 101](#).

#### Procedura

- Per aggiungere un gestore code remoto specificando un CCDT esistente:
  - a) Dalla home page, fare clic su **Connetti gestore code remoto**.
  - b) Specificare il nome del gestore code remoto.
  - c) Facoltativamente, specificare un nome univoco per il gestore code. Se non si specifica un nome univoco, il nome effettivo viene utilizzato con il prefisso "remote-" aggiunto.
  - d) Assicurati che sia selezionato **Connect using a JSON CCDT**.
  - e) Fare clic su **Sfoggia** e selezionare il file contenente la CCDT JSON che si desidera utilizzare.
  - f) Fare clic su **Avanti** per passare alla pagina utente e, facoltativamente, specificare un nome utente e una password per connettersi al gestore code remoto. Se non si specificano queste informazioni, le informazioni di autenticazione vengono prese dal file di configurazione della connessione remota.
  - g) Fare clic su **Avanti** per passare alla pagina Certificato. Se CCDT specifica le informazioni "transmissionSecurity", vengono utilizzate queste informazioni. Puoi facoltativamente incollare un certificato (come chiave pubblica codificata base64) e questo viene aggiunto al truststore globale. Il certificato viene memorizzato temporaneamente in `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` prima di essere aggiunto al truststore. Quando la connessione viene aggiunta correttamente, il certificato viene eliminato da questa ubicazione.

- h) Fare clic su **Avanti** per visualizzare la pagina di riepilogo. È possibile utilizzare il pulsante **Indietro** per rivisitare le pagine precedenti e apportare correzioni. Se si è soddisfatti delle informazioni, fare clic su **Connetti** per connettersi al gestore code remoto.
- Per aggiungere un gestore code remoto e specificare le informazioni di connessione manualmente:
    - a) Dalla home page, fare clic su **Connetti gestore code remoto**.
    - b) Specificare il nome del gestore code remoto.
    - c) Facoltativamente, specificare un nome univoco per il gestore code. Se non si specifica un nome univoco, il nome effettivo viene utilizzato con il prefisso "remote-" aggiunto.
    - d) Selezionare **Immissione manuale**.
    - e) Immettere il nome del canale di connessione client che verrà utilizzato dalla connessione.
    - f) Specificare il nome dell'host su cui è in esecuzione il gestore code remoto. Se vengono rilevate installazioni remote di MQ, vengono visualizzati i nomi host ed è possibile selezionare l'host del gestore code remoto a cui si desidera connettersi. In alcune configurazioni di rete, non è possibile rilevare istanze MQ remote. In questo caso, aggiungere manualmente il nome host e la porta.
    - g) Fare clic su **Avanti** per passare alla pagina utente e, facoltativamente, specificare un nome utente e una password per connettersi al gestore code remoto. Se non si specificano queste informazioni, le informazioni di autenticazione vengono prese dal file di configurazione della connessione remota.
    - h) Fare clic su **Avanti** per passare alla pagina Certificato. È possibile selezionare un CipherSpec SSL dall'elenco a discesa. Puoi facoltativamente incollare un certificato (come chiave pubblica codificata base64) e questo viene aggiunto al truststore globale.
 

Il certificato viene memorizzato temporaneamente in `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` prima di essere aggiunto al truststore. Quando la connessione viene aggiunta correttamente, il certificato viene eliminato da questa ubicazione.
    - i) Fare clic su **Avanti** per visualizzare la pagina di riepilogo. È possibile utilizzare il pulsante **Indietro** per rivisitare le pagine precedenti e apportare correzioni. Se si è soddisfatti delle informazioni, fare clic su **Connetti** per connettersi al gestore code remoto.

Le informazioni di connessione specificate vengono scritte nel file CCDT nella directory web. Il percorso è `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`.

## Risultati

Il gestore code remoto viene visualizzato nell'elenco dei gestori code remoti in IBM MQ Console. Se la connessione ha esito positivo, è possibile gestire gli oggetti del gestore code remoto nello stesso modo in cui si gestiscono gli oggetti di un gestore code locale.

### **V 9.2.3** Connessione di un gestore code remoto a MQ Console utilizzando la riga comandi

È possibile collegare i gestori code remoti a MQ Console utilizzando il comando **setmqweb remote** sulla riga comandi. Un gestore code remoto può essere un gestore code in esecuzione in un'installazione differente sullo stesso sistema di MQ Console oppure un gestore code in esecuzione su un sistema differente.

## Prima di iniziare

- Assicurarsi che il server mqweb sia configurato per consentire le connessioni del gestore code remoto a MQ Console. Per ulteriori informazioni, consultare [Configurazione del comportamento delle connessioni del gestore code remoto](#).

## Procedura

1. Configurare il gestore code locale QML per accettare connessioni remote:
  - a) Creare un canale di connessione server per consentire la gestione remota del gestore code.

È possibile utilizzare MQ Console per creare il canale di connessione server oppure è possibile utilizzare il comando MQSC **DEFINE CHANNEL** sulla riga comandi.

Ad esempio, per creare un canale di connessione server QM1 . SVRCONN per il gestore code remoto QM1, immettere i seguenti comandi:

```
runmqsc QM1
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

Per ulteriori informazioni su **DEFINE CHANNEL** e sulle opzioni disponibili, consultare [DEFINE CHANNEL](#).

- b) Assicurarsi che il canale di connessione server consenta un accesso ID utente appropriato al canale. Questo ID utente deve essere l'ID utente con cui viene avviato il server mqweb sul sistema su cui MQ Console effettua la connessione remota.

È possibile utilizzare MQ Console per creare il record di autorizzazione appropriato oppure è possibile utilizzare il comando MQSC **SET CHLAUTH** sulla riga comandi.

Ad esempio, per autorizzare un utente, `exampleUser`, ad accedere a QM1 . SVRCONN per il gestore code remoto QM1, immettere il seguente comando:

```
SET CHLAUTH(QM1.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') MCAUSER('exampleUser')
```

In questo esempio, il parametro **address** è impostato in modo che `exampleUser` possa connettersi da qualsiasi indirizzo IP. È possibile impostare il parametro **address** su un indirizzo IP specifico. Ad esempio, è possibile limitare l'accesso solo all'indirizzo IP da cui MQ Console si connette in remoto al gestore code. Per ulteriori informazioni sulle opzioni disponibili per questo comando, consultare [SET CHLAUTH](#).

c) 

Creare un listener per accettare le connessioni di rete in entrata:

È possibile utilizzare MQ Console per creare un listener oppure è possibile utilizzare il comando MQSC **DEFINE LISTENER** sulla riga comandi.

Ad esempio, per creare un listener REMOTE . LISTENER sulla porta 1414 per il gestore code QM1, immettere il seguente comando:

```
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
```

- d) Verificare che il listener sia in esecuzione:

È possibile utilizzare MQ Console per avviare il listener oppure è possibile utilizzare il comando MQSC **START LISTENER** sulla riga comandi.

 Ad esempio, su AIX, Linux, and Windows per avviare il listener REMOTE . LISTENER per il gestore code QM1, immettere il seguente comando:

```
START LISTENER(REMOTE.LISTENER)
```

 Ad esempio, su z/OS, per avviare il listener, immettere il comando seguente:

```
START LISTENER TRPTYPE(TCP) PORT(1414)
```

Tenere presente che lo spazio di indirizzo dell'iniziatore di canali deve essere avviato prima di poter avviare un listener su z/OS.

2. Creare un file CCDT JSON che contiene informazioni sulla connessione del gestore code remoto:

- Generare un file CCDT da una definizione di gestore code locale utilizzando il MQ Console associato alla stessa installazione del gestore code a cui si desidera connettersi in remoto.

Dal pannello **Home** , fare clic sul riquadro **Scarica file di connessione** .

- Crea un file CCDT in formato JSON che definisce la connessione. Per ulteriori informazioni sulla creazione di una CCDT in formato JSON, consultare [Configurazione di una CCDT in formato JSON](#).

Il file CCDT deve includere le informazioni `name`, `clientConnection` e `type`. Il file CCDT deve contenere i dati di connessione per il gestore code remoto (QM1) e deve essere memorizzato nell'host del gestore code locale (QML), lo stesso host che ha MQ Console. In sintesi, spetta al gestore code locale QML connettersi al gestore code remoto QM1.

Facoltativamente, è possibile includere ulteriori informazioni come le informazioni `transmissionSecurity`. Per ulteriori informazioni su tutte le definizioni di attributo del canale CCDT, consultare [Elenco completo delle definizioni di attributo del canale CCDT](#).

Il seguente esempio mostra un file CCDT JSON di base per una connessione del gestore code remoto. Imposta il nome del canale sullo stesso nome del canale di collegamento server di esempio creato nel passo ["1" a pagina 101](#) e la porta di collegamento sullo stesso valore della porta utilizzata dal listener. L'host di connessione è impostato con il nome host del sistema su cui è in esecuzione il gestore code remoto di esempio, QM1:

```
{
  "channel": [
    {
      "name": "QM1.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "example.com",
            "port": 1414
          }
        ]
      },
      "queueManager": "QM1"
    },
    {
      "transmissionSecurity": {
        "cipherSpecification": "",
        "certificateLabel": "",
        "certificatePeerName": ""
      }
    }
  ],
  "type": "clientConnection"
}
```

3. Utilizzare il comando **setmqweb remote** per aggiungere le informazioni sul gestore code remoto alla configurazione MQ Console. È necessario utilizzare il comando **setmqweb** associato all'installazione da cui si intende visualizzare le informazioni sul gestore code remoto.

Come minimo, per aggiungere un gestore code remoto al MQ Console è necessario fornire il nome del gestore code, un nome univoco per il gestore code (per differenziare tra altri gestori code remoti che potrebbero avere lo stesso nome del gestore code) e l'URL CCDT per il gestore code. Esistono diverse opzioni aggiuntive che è possibile specificare, come il nome utente e la password da utilizzare per la connessione del gestore code remoto o i dettagli del truststore e del keystore. Per un elenco completo dei parametri che è possibile specificare con il comando **setmqweb remote**, consultare [setmqweb](#).

Ad esempio, per aggiungere il gestore code remoto di esempio QM1, utilizzando il file CCDT di esempio e specificando che il nome utente da utilizzare per la connessione è `exampleUser`, immettere il seguente comando:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL
"c:\myccdt\ccdt.json" -username "exampleUser" -password "password"
```

## Risultati

Il gestore code remoto viene visualizzato nell'elenco dei gestori code remoti in IBM MQ Console. Se la connessione ha esito positivo, è possibile gestire gli oggetti del gestore code remoto nello stesso modo in cui si gestiscono gli oggetti di un gestore code locale.

## V 9.2.0 Utilizzo degli oggetti IBM MQ

A ciascun gestore code IBM MQ sono associati diversi tipi di oggetti.

### Informazioni su questa attività

È possibile utilizzare la console per gestire i seguenti tipi di oggetto IBM MQ :

- Code
- Argomenti
- Sottoscrizione
- Oggetti di comunicazione:
  - Listener
  - Canali gestore code
  - Canali app

### Procedura

Per gestire un oggetto IBM MQ :

1. Nella vista elenco dei gestori code, selezionare il gestore code proprietario degli oggetti che si desidera gestire.
2. Fare clic sulla scheda Code, Argomenti, Sottoscrizioni o Comunicazione per selezionare il tipo di oggetto che si desidera utilizzare.
3. Consultare uno dei seguenti argomenti per istruzioni dettagliate sulla gestione di quel tipo di oggetto.

## V 9.2.0 Gestione delle code

È possibile visualizzare le code esistenti per un determinato gestore code nella scheda **Code** . È possibile aggiungere ed eliminare code, aggiungere e cancellare messaggi da una coda, sfogliare messaggi, visualizzare e impostare le proprietà di una coda e gestire i record di autorizzazioni di una coda.

### Informazioni su questa attività

La vista delle code elenca le code che esistono per un gestore code specifico. Si accede all'elenco di code facendo clic su un gestore code e selezionando la scheda **Code** . È possibile selezionare singole code dall'elenco da gestire.

 Non è possibile visualizzare o modificare i record di autorizzazione per le code su z/OS.

### Procedura

- Per aggiungere una coda:

- a) Nella scheda **Code** , fare clic sul pulsante  .
- b) Selezionare il tipo di coda che si desidera creare:
  - Coda locale - memorizza i messaggi nel gestore code a cui appartiene.
  - Coda alias - un puntatore a un'altra coda sullo stesso gestore code.
  - Coda remota - un puntatore a un'altra coda su un gestore code differente.
  - Coda modello - un modello per una coda utilizzato quando viene creato un gestore code dinamico.
- c) Fornire le informazioni richieste per il tipo di coda che si sta creando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.

- d) Fai clic su **Crea**. Viene creata la nuova coda.
- Per inserire i messaggi in una coda:
  - a) Fare clic sulla coda a cui si desidera aggiungere i messaggi nell'elenco nella vista elenco code. Non è possibile selezionare una coda modello.



- b) Fare clic sul pulsante Crea
- c) Immettere il messaggio che si desidera inserire nella coda.
- d) Fai clic su **Crea**.
- Per cancellare i messaggi da una coda:
  - a) Fare clic sulla coda locale da cui si desidera cancellare i messaggi dall'elenco delle code.



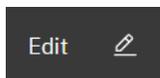
- b) Fare clic sul pulsante Azioni e selezionare **Cancella messaggi**.
- c) Confermare che si desidera cancellare la coda facendo clic su **Cancella messaggi**.
- Per sfogliare i messaggi su una coda, fare clic sulla coda nella vista elenco delle code. Viene visualizzato un elenco dei messaggi su tale coda.

- Per eliminare una coda:

- a) Fare clic sull'icona della chiave  accanto alla coda che si desidera eliminare.
- b) Nella vista Modifica coda, fare clic su **Elimina coda**.
- c) Confermare che si desidera eliminare la coda facendo clic su **Elimina**. La coda è stata eliminata.

- Per visualizzare e modificare le proprietà di una coda:

- a) Fare clic sull'icona della chiave  accanto alla coda che si desidera modificare.



- b) Fare clic sul pulsante Modifica
- c) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi. Per informazioni sulle proprietà, consultare [Proprietà delle code](#) nella MQ Explorer
- d) Fare clic su **Salva** per salvare le modifiche.
- Per visualizzare e modificare i record di autorizzazione per una coda:
  - a) Fare clic sull'icona della chiave  accanto alla coda per cui si desidera modificare i record di autorizzazione.
  - b) Selezionare la scheda **Sicurezza**.
  - c) Gestire i record di autorizzazioni come descritto per i record di autorizzazioni del gestore code. Consultare [“Gestione dei record di autorizzazione del gestore code”](#) a pagina 98.

## **V 9.2.0** *Utilizzo degli argomenti*

È possibile utilizzare IBM MQ Console per aggiungere ed eliminare argomenti e visualizzare e impostare le proprietà di un argomento.

### **Informazioni su questa attività**

La vista degli argomenti elenca gli argomenti esistenti per un gestore code specifico. Accedere all'elenco di argomenti facendo clic su un gestore code e selezionando la scheda **Argomenti**. È possibile selezionare singoli argomenti dall'elenco da utilizzare.

 Non è possibile visualizzare o modificare i record di autorizzazioni per un argomento su z/OS.

## Procedura

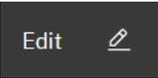
- Per aggiungere un argomento:

- a) Nella scheda **Argomenti**, fare clic sul pulsante Crea .
- b) Fornire le informazioni richieste per l'argomento che si sta creando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.
- c) Fai clic su **Crea**. Viene creato il nuovo argomento.

- Per eliminare un argomento:

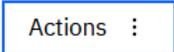
- a) Fare clic sull'icona della chiave  accanto all'argomento che si desidera eliminare.
- b) Nella vista Modifica coda, fare clic su **Elimina argomento**.
- c) Confermare che si desidera eliminare l'argomento facendo clic su **Elimina**. L'argomento viene eliminato.

- Per visualizzare e modificare le proprietà di un argomento:

- a) Fare clic sull'icona della chiave  accanto all'argomento che si desidera modificare.
- b) Fare clic sul pulsante Modifica .
- c) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi. Per informazioni sulle proprietà, consultare [Proprietà degli argomenti](#) nella documentazione di MQ Explorer.
- d) Fare clic su **Salva** per salvare le modifiche.

- Per pubblicare un messaggio su un argomento, è necessario avere almeno una sottoscrizione corrispondente. È possibile creare una sottoscrizione di test, se necessario:

- a) Fare clic sull'argomento che si desidera pubblicare nell'elenco di argomenti.
- b) L'azione successiva dipende dal fatto che questo argomento abbia sottoscrizioni corrispondenti:  
Se non ci sono sottoscrizioni corrispondenti:

- a. Fare clic sul pulsante Azioni  e selezionare **Argomento di test**.

- b. Fare clic sul pulsante Test . Un messaggio di test viene scritto in una sottoscrizione di test.

Se sono presenti sottoscrizioni corrispondenti per l'argomento:

- a. Fare clic sul nome di una sottoscrizione.

- b. Fare clic sul pulsante Crea .
- c. Immettere il messaggio che si desidera pubblicare.



d. Fare clic sul pulsante Inserisci . Il messaggio viene scritto in tutte le sottoscrizioni corrispondenti.

- Per sottoscrivere un argomento, consultare [“Utilizzo delle sottoscrizioni”](#) a pagina 107:
- Per visualizzare e modificare i record di autorizzazione per un argomento:
  - a) Fare clic sull'icona della chiave  accanto all'argomento per cui si desidera modificare i record di autorizzazione.
  - b) Selezionare la scheda **Sicurezza**.
  - c) Gestire i record di autorizzazione come descritto per i record di autorizzazione del gestore code, consultare [“Gestione dei record di autorizzazione del gestore code”](#) a pagina 98.

## **Utilizzo delle sottoscrizioni**

È possibile utilizzare IBM MQ Console per aggiungere ed eliminare le sottoscrizioni e visualizzare e impostare le proprietà di una sottoscrizione.

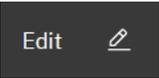
### Informazioni su questa attività

La vista Sottoscrizioni elenca le sottoscrizioni che esistono per un determinato gestore code. Per accedere all'elenco delle sottoscrizioni, fare clic su un gestore code e selezionare la scheda **Sottoscrizioni**. È possibile selezionare singole sottoscrizioni dall'elenco da utilizzare.

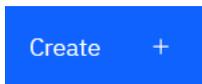
Per ulteriori informazioni sulle sottoscrizioni, consultare [Sottoscrittori e sottoscrizioni](#) e [DEFINE SUB](#).

 Non è possibile visualizzare o modificare i record di autorizzazione per una sottoscrizione su z/OS.

### Procedura

- Per aggiungere una sottoscrizione:
  - a) Nella scheda **Sottoscrizioni**, fare clic sul pulsante Crea .
  - b) Scegliere se si desidera creare una sottoscrizione gestita o non gestita.
  - c) Fornire le informazioni richieste per la sottoscrizione che si sta creando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.
  - d) Fai clic su **Crea**. Viene creata la nuova sottoscrizione.
- Per eliminare una sottoscrizione:
  - a) Fare clic sull'icona della chiave  accanto alla sottoscrizione che si desidera eliminare.
  - b) Nella vista Modifica coda, fare clic su **Elimina sottoscrizione**.
  - c) Confermare che si desidera eliminare la sottoscrizione facendo clic su **Elimina**. La sottoscrizione viene eliminata.
- Per visualizzare e modificare le proprietà di una sottoscrizione:
  - a) Fare clic sull'icona della chiave  accanto alla sottoscrizione che si desidera modificare.
  - b) Fare clic sul pulsante Modifica .

- c) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi.
- d) Fare clic su **Salva** per salvare le modifiche.
- Per pubblicare un messaggio sull'argomento a cui è sottoscritta la sottoscrizione:
  - a) Fare clic sulla sottoscrizione di cui si desidera pubblicare l'argomento nell'elenco di sottoscrizioni.



- b) Fare clic sul pulsante Crea.
- c) Immettere il messaggio che si desidera pubblicare.



- d) Fare clic sul pulsante Inserisci. Il messaggio viene scritto in tutte le sottoscrizioni che corrispondono all'argomento pubblicato.

### V 9.2.0 Utilizzo dei listener

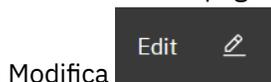
È possibile utilizzare IBM MQ Console per aggiungere ed eliminare i listener, avviare e arrestare i listener, visualizzare e impostare le proprietà del listener e gestire i record di autorizzazioni per un listener.

## Informazioni su questa attività

La vista dei listener visualizza i listener che esistono per un gestore code specifico. È possibile selezionare singoli listener da utilizzare.

## Procedura

- Per creare un listener:
  - a) Nella scheda **Comunicazione**, verificare che sia visualizzata la vista dei listener e fare clic sul pulsante Crea.
- b) Fornire le informazioni richieste per il listener che si sta creando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.
- c) Fai clic su **Crea**. Il nuovo listener viene creato.
- Per avviare un listener:
  - a) Individuare il listener che si desidera avviare nell'elenco.
  - b) Selezionare **Avvia** dal menu .
- Per arrestare un listener:
  - a) Individuare il listener che si desidera avviare nell'elenco.
  - b) Selezionare **Arresta** dal menu .
- Per visualizzare e modificare le proprietà di un listener:
  - a) Individuare il listener nell'elenco.
  - b) Selezionare **Configura** dal menu .
  - c) Verificare che la pagina **Proprietà** sia selezionata. Per modificare le proprietà, fare clic sul pulsante



- d) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi. Per ulteriori informazioni sulle proprietà, consultare [Proprietà del listener](#) nella documentazione di MQ Explorer.
- e) Fare clic su **Salva** per salvare le modifiche.
- Per visualizzare e modificare i record di autorizzazioni per un listener:
  - a) Individuare il listener nell'elenco.
  - b) Selezionare **Configura** dal menu .
  - c) Selezionare la scheda **Sicurezza**.
  - d) Gestire i record di autorizzazione come descritto per i record di autorizzazione del gestore code, consultare [“Gestione dei record di autorizzazione del gestore code” a pagina 98](#).
- Per eliminare un listener:
  - a) Individuare il listener nell'elenco.
  - b) Selezionare **Configura** dal menu .
  - c) Fare clic su **Elimina listener**.

### **Utilizzo dei canali del gestore code**

È possibile utilizzare le operazioni IBM MQ Console con i canali del gestore code: è possibile aggiungere ed eliminare canali del gestore code, avviare e arrestare canali, reimpostare e risolvere canali e eseguire il ping dei canali. È inoltre possibile visualizzare e impostare le proprietà di un canale del gestore code e gestire i record di autorizzazioni per il canale.

### **Informazioni su questa attività**

Un canale gestore code è un link di comunicazione logica per la trasmissione dei messaggi tra i gestori code in una rete. La vista del canale del gestore code include un pannello che mostra una vista rapida del numero di canali in esecuzione, del numero di nuovi tentativi e del numero di arrestati.

 Non è possibile visualizzare o modificare i record di autorizzazione per un canale su z/OS.

### **Procedura**

- Per aggiungere un canale del gestore code:
  - a) Nella scheda **Comunicazione**, accertarsi che sia visualizzata la vista dei canali del gestore code e fare clic sul pulsante di creazione .
  - b) Selezionare il tipo di canale del gestore code che si desidera creare e fare clic sul pulsante Avanti .
  - c) Fornire le informazioni richieste per il canale che si sta creando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.
  - d) Fai clic su **Crea**. Il nuovo canale viene creato con lo stato **inactive**.
- Per avviare un canale del gestore code:
  - a) Individuare il canale che si desidera avviare nell'elenco.
  - b) Selezionare **Avvia** dal menu .
- Per arrestare un canale del gestore code:

- a) Individuare il canale che si desidera arrestare nell'elenco.
- b) Selezionare **Arresta** dal menu  .
- Per visualizzare le proprietà di un canale del gestore code:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Configura** dal menu  .
  - c) Verificare che la pagina **Proprietà** sia selezionata. Per modificare le proprietà, fare clic sul pulsante
 

Modifica 
  - d) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi. Per ulteriori informazioni sulle proprietà, consultare [Proprietà del canale](#) nella documentazione di MQ Explorer.
  - e) Fare clic su **Salva** per salvare le modifiche.
- Per reimpostare un canale del gestore code:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Avanzate** dal menu  .
  - c) Nella sezione **Reimposta** , specificare un numero di sequenza del messaggio.  
È necessario reimpostare un canale se non si avvia perché le due estremità non sono d'accordo sul numero di sequenza del messaggio successivo da inviare. Il numero di sequenza del messaggio specifica tale numero.
  - d) Fare clic su **Reimposta canale**.
- Per risolvere un canale mittente o server:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Avanzate** dal menu  .
  - c) Nella sezione **Risolvi** , scegliere se eseguire il commit o il backout del batch corrente di messaggi facendo clic su **Ripristina messaggi nella coda di trasmissione** o **Elimina messaggi**.
- Per eseguire il ping di un canale del gestore code:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Ping** dal menu  .
- Per visualizzare e modificare i record di autorizzazione per un canale del gestore code:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Configura** dal menu  .
  - c) Selezionare la scheda **Sicurezza**.
  - d) Gestire i record di autorizzazione come descritto per i record di autorizzazione del gestore code, consultare [“Gestione dei record di autorizzazione del gestore code”](#) a pagina 98.
- Per eliminare un canale del gestore code:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Configura** dal menu  .
  - c) Fare clic su **Elimina canale**.

## V 9.2.0 Utilizzo dei canali dell'applicazione

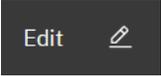
È possibile utilizzare IBM MQ Console per gestire i canali dell'applicazione: è possibile aggiungere ed eliminare canali, avviare e arrestare canali, ripristinare e risolvere canali e eseguire il ping dei canali. È inoltre possibile visualizzare e impostare le proprietà di un canale dell'applicazione e gestire i record di autorizzazioni per il canale.

### Informazioni su questa attività

Un canale applicazione è un link di comunicazione logica, utilizzato dalle applicazioni per connettersi a un gestore code in una rete. La vista del canale dell'applicazione include un pannello che mostra una vista rapida del numero di canali in esecuzione, del numero di nuovi tentativi e del numero di arrestati.

**z/OS** Non è possibile visualizzare o modificare i record di autorizzazione per un canale su z/OS.

### Procedura

- Per aggiungere un canale dell'applicazione:
  - a) Nella scheda **Comunicazione**, assicurarsi che la visualizzazione dei canali dell'app venga visualizzata e fare clic sul pulsante **Crea** .
  - b) Fare clic sul pulsante **Avanti** .
  - c) Fornire le informazioni richieste per il canale che si sta creando. Per impostazione predefinita, vengono visualizzate le proprietà minime consigliate per cui fornire i valori. È possibile visualizzare tutte le proprietà disponibili selezionando **Mostra tutte le opzioni disponibili**.
  - d) Fai clic su **Crea**. Il nuovo canale viene creato con lo stato **inactive**.
- Per avviare un canale dell'applicazione:
  - a) Individuare il canale che si desidera avviare nell'elenco.
  - b) Selezionare **Avvia** dal menu .
- Per arrestare un canale dell'applicazione:
  - a) Individuare il canale che si desidera arrestare nell'elenco.
  - b) Selezionare **Arresta** dal menu .
- Per visualizzare le proprietà di un canale dell'applicazione:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Configura** dal menu .
  - c) Verificare che la pagina **Proprietà** sia selezionata. Per modificare le proprietà, fare clic sul pulsante **Modifica** .
  - d) Modificare le proprietà come desiderato. Se la casella di testo della proprietà è disabilitata, la proprietà è di sola lettura o può essere modificata solo dalla riga comandi. Per ulteriori informazioni sulle proprietà, consultare [Proprietà del canale](#) nella documentazione di MQ Explorer.
  - e) Fare clic su **Salva** per salvare le modifiche.
- Per reimpostare un canale dell'applicazione:
  - a) Individuare il canale nell'elenco.

- b) Selezionare **Avanzate** dal menu  .
- c) Nella sezione **Reimposta** , specificare un numero di sequenza del messaggio.  
È necessario reimpostare un canale se non si avvia perché le due estremità non sono d'accordo sul numero di sequenza del messaggio successivo da inviare. Il numero di sequenza del messaggio specifica tale numero.
- d) Fare clic su **Reimposta canale**.
- Per risolvere un canale mittente o server:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Avanzate** dal menu  .
  - c) Nella sezione **Risolvi** , scegliere se eseguire il commit o il backout del batch corrente di messaggi facendo clic su **Ripristina messaggi nella coda di trasmissione** o **Elimina messaggi**.
- Per emettere un comando ping per un canale:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Ping** dal menu  .
- Per visualizzare e modificare i record di autorizzazione per un canale dell'applicazione:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Configura** dal menu  .
  - c) Selezionare la scheda **Sicurezza**.
  - d) Gestire i record di autorizzazione come descritto per i record di autorizzazione del gestore code, consultare [“Gestione dei record di autorizzazione del gestore code”](#) a pagina 98.
- Per eliminare un canale dell'applicazione:
  - a) Individuare il canale nell'elenco.
  - b) Selezionare **Configura** dal menu  .
  - c) Fare clic su **Elimina canale**.

## **Impostazioni della console Web**

È possibile specificare alcune impostazioni generali per la console Web.

Fare clic sull'icona Impostazioni  **Settings** per passare alla vista delle impostazioni della console Web.

Utilizzare le impostazioni per controllare le funzioni seguenti:

- Aggiornamento automatico dei gestori code ogni 10 secondi. Questa funzione può essere attivata o disattivata.
- Indica se gli oggetti di sistema vengono visualizzati. È possibile specificare questo valore per tutti i tipi di oggetto oppure selezionare i tipi di oggetto singolarmente.
- Se le informazioni di traccia vengono raccolte o meno.

## **Passaggio da un tipo di console all'altro**

È possibile passare da New Web Console (che è la console Web predefinita per IBM MQ 9.2) a Dashboard Web Console (che era la console Web per le versioni precedenti di IBM MQ).

## Informazioni su questa attività

Si passa da una console all'altra utilizzando il comando **setmqweb**.

Se si torna all'utilizzo di Dashboard Web Console, consultare [The Dashboard Web Console](#) nella documentazione di IBM MQ 9.1 per istruzioni sull'utilizzo.

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmqweb** e **setmqweb**:

- ▶ **z/OS** Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmqweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
- ▶ **Multi** Su tutti gli altri sistemi operativi, è necessario essere un [utente privilegiato](#).



### Attenzione: ▶ **z/OS**

Prima di immettere i comandi **setmqweb** o **dspmqweb** in z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR`, in modo che la variabile punti alla configurazione del server mqweb.

Per eseguire questa operazione, immettere il seguente comando:

```
export WLP_USER_DIR=WLP_user_directory
```

dove `WLP_user_directory` è il nome della directory passata a `crtmqweb`. Ad esempio:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

Per ulteriori informazioni, consultare [Creazione del server mqweb](#).

## Procedura

- Per passare a Dashboard Web Console, utilizzare i comandi seguenti:
  - Impostare la proprietà `mqConsoleEnableUnsafeInline` `true` immettendo il seguente comando:

```
setmqweb properties -k mqConsoleEnableUnsafeInline -v true
```

Tenere presente che l'impostazione di questa proprietà rende meno sicura la serie di regole CSP per abilitare l'inline non sicura, che potrebbe rendere la configurazione meno sicura, ma è richiesta da Dashboard Web Console

- Passare al server mqweb per utilizzare Dashboard Web Console immettendo il seguente comando:

```
setmqweb properties -k mqConsoleEarName -v com.ibm.mq.console
```

- Per ripristinare le impostazioni predefinite per il server web mq (che include New Web Console), utilizzare il seguente comando:

```
setmqweb properties -r
```

- Per tornare a New Web Console conservando tutte le altre personalizzazioni eseguite sul server mqweb, utilizzare i seguenti comandi:

- Impostare la proprietà in linea `mqConsoleEnableUnsafeInline` `false` per ripristinare la configurazione della sicurezza immettendo il seguente comando:

```
setmqweb properties -k mqConsoleEnableUnsafeInline -v false
```

- Passare al server mqweb per utilizzare New Web Console immettendo il seguente comando:

```
setmqweb properties -k mqConsoleEarName -v com.ibm.mq.webconsole
```

## Riferimenti correlati

[setmqweb](#)

IBM MQ Explorer consente di eseguire la gestione locale o remota della rete solo da un computer su cui è in esecuzione Windows o Linux x86-64 .

IBM MQ for Windows e IBM MQ per Linux x86-64 forniscono un'interfaccia di gestione denominata IBM MQ Explorer per eseguire attività di gestione come alternativa all'utilizzo dei comandi di controllo o MQSC. [Confronto delle serie di comandi](#) mostra le operazioni che è possibile effettuare utilizzando IBM MQ Explorer.

Il IBM MQ Explorer consente di eseguire la gestione locale o remota della rete da un computer su cui è in esecuzione Windows o Linux x86-64, puntando il IBM MQ Explorer sui gestori code e sui cluster a cui si è interessati. Può connettersi in remoto ai gestori code in esecuzione su qualsiasi piattaforma supportata, tra cui z/OS, consentendo di visualizzare, esplorare e modificare l'intera struttura portante di messaggistica dalla console.

Per configurare i gestori code IBM MQ remoti in maniera che IBM MQ Explorer possa gestirli, consultare [“Software prerequisito e definizioni per IBM MQ Explorer”](#) a pagina 116.

Consente di eseguire attività, di solito associate all'impostazione e all'ottimizzazione dell'ambiente di lavoro per IBM MQ, in locale o in remoto all'interno di un dominio di sistema Windows o Linux x86-64 .

Su Linux, l'avvio di IBM MQ Explorer potrebbe non riuscire se si dispone di più di un'installazione di Eclipse . Se ciò si verifica, avviare IBM MQ Explorer utilizzando un ID utente diverso da quello utilizzato per l'altra installazione di Eclipse .

Su Linux, per avviare correttamente IBM MQ Explorer , è necessario essere in grado di scrivere un file nella propria directory home e la directory home deve esistere.

IBM MQ Explorer può essere installato sia come parte dell'installazione del prodotto (consultare [Installazione e disinstallazione di IBM MQ](#)), sia dal download di IBM MQ Explorer autonomo disponibile da Fix Central (consultare [Installazione e disinstallazione di IBM MQ Explorer come applicazione autonoma su Linux e Windows](#)).

È possibile utilizzare IBM MQ Explorer per eseguire attività di gestione utilizzando una serie di finestre di dialogo Proprietà e Viste contenuto. È anche possibile estendere IBM MQ Explorer scrivendo uno o più plugin Eclipse .

### IBM MQ Explorer attività

Con IBM MQ Explorer, è possibile eseguire le seguenti attività:

- Creare ed [eliminare un gestore code](#) (solo sulla macchina locale).
- [Avviare e arrestare un gestore code](#) (solo sulla macchina locale).
- [Definire, visualizzare e modificare le definizioni di IBM MQ oggetti](#) come code e canali.
- [Sfogliare i messaggi su una coda](#).
- [Avviare e arrestare un canale](#).
- [Visualizzare le informazioni di stato](#) su un canale, listener, coda o oggetti di servizio.
- Visualizzare i gestori code in un cluster.
- Verificare quali applicazioni, utenti o canali hanno una particolare coda aperta.
- [Creare un nuovo cluster di gestori code](#) utilizzando la procedura guidata Crea nuovo cluster.
- [Aggiungere un gestore code a un cluster](#) utilizzando la procedura guidata Aggiungi gestore code al cluster.
- [Gestire l'oggetto delle informazioni di autenticazione](#), utilizzato con la sicurezza del canale TLS (Transport Layer Security).
- Creare ed eliminare iniziatori di canali, controlli trigger e listener.

- Avviare o arrestare i server dei comandi [../com.ibm.mq.explorer.doc/e\\_cmdserver\\_starting.dita](#), gli iniziatori di canali, i controlli dei trigger e i listener.
- Impostare specifici servizi su [avvio automatico all'avvio di un gestore code](#).
- Modificare le proprietà dei gestori code.
- Modificare il gestore code predefinito locale.
- Richiamare la **strmqikm** GUI (ikeyman) per gestire certificati TLS, associare certificati ai gestori code e configurare e configurare archivi di certificati (solo sulla macchina locale).
- Creare JMS oggetti da IBM MQ oggetti [IBM MQ oggetti da JMS oggetti](#).
- Creare un JMS factory di connessione per uno qualsiasi dei tipi attualmente supportati.
- Modificare i parametri per qualsiasi servizio, come il numero di porta TCP per un listener o il nome della coda dell'iniziatore di canali.
- [Avviare o arrestare la traccia del servizio](#).

## Finestre di dialogo Proprietà e Viste contenuto

Le attività di gestione vengono eseguite utilizzando una serie di finestre di dialogo Proprietà e Viste contenuto.

### Vista Contenuto

Una vista Contenuto è un pannello che può visualizzare quanto segue:

- Attributi e opzioni di gestione relativi a IBM MQ .
- Attributi e opzioni di gestione relativi a uno o più oggetti correlati.
- Attributi e opzioni di amministrazione per un cluster.

### Finestre delle proprietà

Una finestra di dialogo delle proprietà è un pannello che visualizza gli attributi relativi a un oggetto in una serie di campi, alcuni dei quali possono essere modificati.

Navigare attraverso IBM MQ Explorer utilizzando la vista Navigator . Il Navigator consente di selezionare la vista Contenuto richiesta.

## Estensione di IBM MQ Explorer

IBM MQ Explorer presenta le informazioni in uno stile coerente con quello del framework Eclipse e con le altre applicazioni plug-in supportate da Eclipse .

Tramite l'estensione di IBM MQ Explorer, gli amministratori di sistema hanno la possibilità di personalizzare IBM MQ Explorer per migliorare il modo in cui amministrano IBM MQ.

Per ulteriori informazioni, consultare [Estensione di MQ Explorer](#).

## **Decidere se utilizzare il IBM MQ Explorer**

Quando si decide se utilizzare IBM MQ Explorer durante l'installazione, considerare le informazioni elencate in questo argomento.

È necessario essere consapevoli dei seguenti punti:

### Nomi oggetto

Se si utilizzano nomi minuscoli per i gestori code e altri oggetti con IBM MQ Explorer, quando si utilizzano gli oggetti utilizzando i comandi MQSC, è necessario racchiudere i nomi oggetto tra virgolette singole oppure IBM MQ non li riconosce.

### Gestori code di grandi dimensioni

Il IBM MQ Explorer funziona meglio con i gestori code di piccole dimensioni. Se si dispone di un numero elevato di oggetti su un singolo gestore code, potrebbero verificarsi dei ritardi mentre IBM MQ Explorer estrae le informazioni richieste da presentare in una vista.

## Cluster

I cluster IBM MQ possono potenzialmente contenere centinaia o migliaia di gestori code. Il IBM MQ Explorer presenta i gestori code in un cluster utilizzando una struttura ad albero. La dimensione fisica di un cluster non influenza la velocità di IBM MQ Explorer in modo significativo perché IBM MQ Explorer non si connette ai gestori code nel cluster finché non vengono selezionati.

## Impostazione di IBM MQ Explorer

Questa sezione illustra i passi che è necessario eseguire per configurare IBM MQ Explorer.

- [“Software prerequisito e definizioni per IBM MQ Explorer” a pagina 116](#)
- [“Sicurezza per IBM MQ Explorer” a pagina 116](#)
- [“Visualizzazione e nascondimento di gestori code e cluster in IBM MQ Explorer” a pagina 120](#)
- [“Appartenenza al cluster e IBM MQ Explorer” a pagina 121](#)
- [“Conversione dati per IBM MQ Explorer” a pagina 121](#)

## Software prerequisito e definizioni per IBM MQ Explorer

Assicurarsi di soddisfare i seguenti requisiti prima di provare a utilizzare IBM MQ Explorer.

IBM MQ Explorer può connettersi ai gestori code remoti utilizzando solo il protocollo di comunicazione TCP/IP.

Verificare che:

1. Un server dei comandi è in esecuzione su ogni gestore code gestito in remoto.
2. Un oggetto listener TCP/IP adatto deve essere in esecuzione su ogni gestore code remoto. Questo oggetto può essere il listener IBM MQ o, su sistemi AIX and Linux , il daemon inetd.
3. Un canale di connessione server, per impostazione predefinita denominato SYSTEM.ADMIN.SVRCONN, esiste su tutti i gestori code remoti.

È possibile creare il canale utilizzando il seguente comando MQSC:

```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

Questo comando crea una definizione di canale di base. Se si desidera una definizione più sofisticata (per impostare la sicurezza, ad esempio), sono necessari ulteriori parametri. Per ulteriori informazioni, vedere [DEFINE CHANNEL](#).

4. La coda di sistema, SYSTEM.MQEXPLORER.REPLY.MODEL, deve esistere.

## Sicurezza per IBM MQ Explorer

Se si utilizza IBM MQ in un ambiente in cui è importante controllare l'accesso utente a determinati oggetti, potrebbe essere necessario considerare gli aspetti di sicurezza dell'utilizzo di IBM MQ Explorer.

### **Autorizzazione all'utilizzo di IBM MQ Explorer**

Qualsiasi utente può utilizzare IBM MQ Explorer, ma sono necessarie determinate autorizzazioni per connettersi, accedere e gestire i gestori code.

Per eseguire le attività amministrative locali utilizzando IBM MQ Explorer, è necessario che un utente disponga dell'autorizzazione necessaria per eseguire le attività amministrative. Se l'utente è un membro del gruppo mqm , dispone dell'autorizzazione per eseguire tutte le attività amministrative locali.

Per collegarsi a un gestore code remoto ed eseguire attività di gestione remota utilizzando IBM MQ Explorer, all'utente che esegue IBM MQ Explorer è richiesto di disporre delle seguenti autorizzazioni:

- Autorizzazione CONNECT sull'oggetto gestore code di destinazione
- Autorizzazione INQUIRE sull'oggetto del gestore code di destinazione
- Autorizzazione DISPLAY per l'oggetto gestore code di destinazione

- Autorizzazione INQUIRE per la coda, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autorizzazione DISPLAY alla coda, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autorizzazione INPUT (get) per la coda, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autorizzazione OUTPUT (inserimento) per la coda, SYSTEM.MQEXPLORER.REPLY.MODEL
- Autorizzazione OUTPUT (inserimento) per la coda, SYSTEM.ADMIN.COMMAND.QUEUE
- Autorizzazione INQUIRE sulla coda, SYSTEM.ADMIN.COMMAND.QUEUE
- Autorizzazione per eseguire l'azione selezionata

**Nota:** L'autorizzazione INPUT è relativa all'input per l'utente da una coda (un'operazione get). L'autorizzazione OUTPUT si riferisce all'output dall'utente ad una coda (un'operazione di inserimento).

Per connettersi a un gestore code remoto su IBM MQ for z/OS ed eseguire attività di gestione remota utilizzando IBM MQ Explorer, è necessario fornire quanto segue:

- Un profilo RACF per la coda di sistema, SYSTEM.MQEXPLORER.REPLY.MODEL
- Un profilo RACF per le code, AMQ.MQEXPLORER.\*

Inoltre, l'utente che esegue IBM MQ Explorer deve disporre delle seguenti autorizzazioni:

- Autorizzazione RACF UPDATE per la coda di sistema, SYSTEM.MQEXPLORER.REPLY.MODEL
- RACF Autorizzazione UPDATE per le code, AMQ.MQEXPLORER.\*
- Autorizzazione CONNECT sull'oggetto gestore code di destinazione
- Autorizzazione per eseguire l'azione selezionata
- Autorizzazione READ per tutti i profili hlq.DISPLAY.object nella classe MQCMDS

Per informazioni su come concedere l'autorizzazione agli oggetti IBM MQ , consultare [Concessione dell'accesso a un oggetto IBM MQ su sistemi AIX, Linux, and Windows](#).

Se un utente tenta di eseguire un'operazione che non è autorizzato ad eseguire, il gestore code di destinazione richiama le procedure di errore di autorizzazione e l'operazione non riesce.

Il filtro predefinito in IBM MQ Explorer è visualizzare tutti gli oggetti IBM MQ . Se vi sono oggetti IBM MQ per cui un utente non dispone dell'autorizzazione DISPLAY, vengono generati errori di autorizzazione. Se si stanno registrando degli eventi di autorizzazione, limitare l'intervallo di oggetti visualizzati agli oggetti per i quali l'utente dispone dell'autorizzazione DISPLAY.

### ***Sicurezza per la connessione a gestori code remoti da IBM MQ Explorer***

È necessario proteggere il canale tra IBM MQ Explorer e ciascun gestore code remoto.

IBM MQ Explorer si collega ai gestori code remoti come applicazione client MQI. Ciò significa che ogni gestore code remoto deve avere una definizione di un canale di connessione server e un listener TCP/IP adatto. Se non si protegge il canale di connessione server, è possibile che un'applicazione dannosa si connetta allo stesso canale di connessione server e ottenga l'accesso agli oggetti del gestore code con autorizzazione illimitata. Per proteggere il canale di connessione del server, specificare un valore non vuoto per l'attributo MCAUSER del canale, utilizzare i record di autenticazione del canale o utilizzare un'uscita di sicurezza.

**Il valore predefinito dell'attributo MCAUSER è l'ID utente locale.** Se si specifica un nome utente non vuoto come attributo MCAUSER del canale di connessione del server, tutti i programmi che si collegano al gestore code utilizzando questo canale vengono eseguiti con l'identità dell'utente denominato e dispongono dello stesso livello di autorizzazione. Ciò non si verifica se si utilizzano i record di autenticazione di canale.

### ***Utilizzo di un'uscita di sicurezza con IBM MQ Explorer***

È possibile specificare un'uscita di sicurezza predefinita e uscite di sicurezza specifiche del gestore code utilizzando IBM MQ Explorer.

È possibile definire un'uscita di sicurezza predefinita, che può essere utilizzata per tutte le connessioni client nuove da IBM MQ Explorer. Questa uscita predefinita può essere sovrascritta al momento della

connessione. È anche possibile definire un'uscita di sicurezza per un singolo gestore code o una serie di gestori code, che diventa effettiva quando viene effettuata una connessione. Le uscite vengono specificate utilizzando IBM MQ Explorer. Per ulteriori informazioni, vedere l' Aiuto di IBM MQ Explorer.

### **Utilizzo di IBM MQ Explorer per la connessione a un gestore code remoto utilizzando i canali MQI abilitati a TLS**

IBM MQ Explorer si connette ai gestori code remoti utilizzando un canale MQI. Se si desidera proteggere il canale MQI utilizzando la sicurezza TLS, è necessario stabilire il canale utilizzando una tabella di definizione del canale client.

Per informazioni su come stabilire un canale MQI utilizzando una tabella di definizione di canale client, consultare [Panoramica di IBM MQ MQI clients](#).

Una volta stabilito il canale utilizzando una tabella di definizione del canale del client, è possibile utilizzare IBM MQ Explorer per connettersi a un gestore code remoto utilizzando il canale MQI abilitato a TLS, come descritto in [“Attività sul sistema che ospita il gestore code remoto”](#) a pagina 118 e [“Attività sul sistema che ospita IBM MQ Explorer”](#) a pagina 118.

### **Attività sul sistema che ospita il gestore code remoto**

Sul sistema che ospita il gestore code remoto, effettuare le seguenti attività:

1. Definire una coppia di canali di connessione server e client e specificare il valore appropriato per l'attributo `SSLCIPH` sulla connessione server su entrambi i canali. Per ulteriori informazioni sull'attributo `SSLCIPH`, consultare [Protezione dei canali con TLS](#).
2. Inviare la tabella di definizione del canale `AMQCLCHL.TAB`, che si trova nella directory `@ipcc` del gestore code, al sistema su cui è presente IBM MQ Explorer.
3. Avviare un listener TCP/IP su una porta designata.
4. Inserire i certificati CA e TLS personali nella directory SSL del gestore code:
  -  `/var/mqm/qmgrs/+QMNAME+/SSL` per sistemi AIX o Linux .
  -  `C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL` per sistemi Windows .Dove `+QMNAME+` è un token che rappresenta il nome del gestore code.
5. Creare un file database delle chiavi di tipo CMS denominato `key.kdb`. Memorizzare la password in un file selezionando l'opzione in `strmqikm` (iKeyman) GUI o utilizzando l'opzione `-stash` con il comando `runmqckm` o il comando `runmqakm`.
6. Aggiungere i certificati CA al database delle chiavi creato nel passo precedente.
7. Importare il certificato personale per il gestore code nel database di chiavi.

Per informazioni più dettagliate sull'utilizzo di TLS su sistemi Windows, consulta [Utilizzo di TLS su AIX, Linux, and Windows](#).

### **Attività sul sistema che ospita IBM MQ Explorer**

Sul sistema su cui è presente IBM MQ Explorer, effettuare le seguenti attività:

1. Creare un file database di chiavi di tipo JKS denominato `key.jks`. Impostare una password per questo file di database delle chiavi.

Il IBM MQ Explorer utilizza i file keystore Java (JKS) per la sicurezza TLS e quindi il file keystore creato per configurare TLS per il IBM MQ Explorer deve corrispondere a questo.
2. Aggiungere i certificati CA al database delle chiavi creato nel passo precedente.
3. Importare il certificato personale per il gestore code nel database di chiavi.
4. Sui sistemi Windows e Linux, avviare IBM MQ Explorer utilizzando il menu di sistema, il file eseguibile `MQExplorer` o il comando `strmqcfig`.

5. Dalla barra degli strumenti di IBM MQ Explorer , fare clic su **Finestra -> Preferenze**, quindi espandere **IBM MQ Explorer** e selezionare **Archivi certificati client SSL**. Immettere il nome e la password per il file JKS creato nel passo 1 di “Attività sul sistema che ospita IBM MQ Explorer” a pagina 118, in Trusted Certificate Store e Personal Certificate Store, quindi fare clic su **OK**.
6. Chiudere la finestra **Preferenze** e fare clic con il tasto destro del mouse su **Gestori code**. Fare clic su **Mostra / nascondi gestori code**, quindi fare clic su **Aggiungi** nella schermata **Mostra / nascondi gestori code** .
7. Immettere il nome del gestore code e selezionare l'opzione **Connetti direttamente** . Fare clic su **Avanti**.
8. Selezionare **Utilizza CCDT (client channel definition table)** e specificare l'ubicazione del file di tabella del canale trasferito dal gestore code remoto nel passo 2 in “Attività sul sistema che ospita il gestore code remoto” a pagina 118 sul sistema su cui si trova il gestore code remoto.
9. Fare clic su **Fine**. È ora possibile accedere al gestore code remoto da IBM MQ Explorer.

### **Connessione tramite un altro gestore code con IBM MQ Explorer**

IBM MQ Explorer consente di connettersi a un gestore code tramite un gestore code intermedio, a cui IBM MQ Explorer è già connesso.

In questo caso, IBM MQ Explorer inserisce i messaggi di comando PCF nel gestore code intermedio, specificando quanto segue:

- Il parametro *ObjectQMgrName* nel descrittore oggetto (MQOD) come nome del gestore code di destinazione. Per ulteriori informazioni sulla risoluzione dei nomi delle code, consultare [Risoluzione dei nomi](#).
- Il parametro *UserIdentifier* nel descrittore del messaggio (MQMD) come *userId* locale.

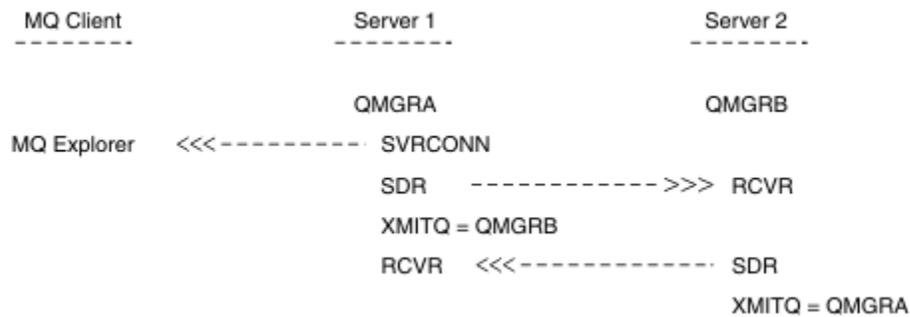
Se la connessione viene quindi utilizzata per connettersi al gestore code di destinazione tramite un gestore code intermedio, l' *userId* viene trasmesso nuovamente nel parametro *UserIdentifier* del descrittore del messaggio (MQMD). Per consentire al listener MCA sul gestore code di destinazione di accettare questo messaggio, è necessario impostare l'attributo MCAUSER oppure è necessario che *userId* esista già con l'autorizzazione di inserimento.

Il server dei comandi sul gestore code di destinazione inserisce i messaggi nella coda di trasmissione specificando l' *userId* nel parametro *UserIdentifier* nel descrittore del messaggio (MQMD). Perché questo inserimento abbia esito positivo, è necessario che l' *userId* esista già sul gestore code di destinazione con autorizzazione di inserimento.

Il seguente esempio mostra come connettere un gestore code, tramite un gestore code intermedio, a IBM MQ Explorer.

Stabilire una connessione di amministrazione remota a un gestore code. Verificare che:

- Il gestore code sul server è attivo e ha un canale di connessione server (SVRCONN) definito.
- Il listener è attivo.
- Il server dei comandi è attivo.
- SYSTEM.MQ EXPLORER.REPLY.MODEL è stata creata e si dispone dell'autorità sufficiente.
- I listener del gestore code, i server dei comandi e i canale mittente vengono avviati.



In questo esempio:

- IBM MQ Explorer è connesso al gestore code QMGRA (in esecuzione su Server1) utilizzando una connessione client.
- Il gestore code QMGRB su Server2 può ora essere connesso a IBM MQ Explorer tramite un gestore code intermedio ( QMGRA)
- Quando ci si connette a QMGRB con IBM MQ Explorer, selezionare QMGRA come gestore code intermedio

In questa situazione, non vi è alcuna connessione diretta a QMGRB da IBM MQ Explorer; la connessione a QMGRB avviene tramite QMGRA.

Il gestore code QMGRB su Server2 si connette a QMGRA su Server1 utilizzando i canali mittente - destinatario. Il canale tra QMGRA e QMGRB deve essere impostato in modo tale che l'amministrazione remota sia possibile; consultare [“Configurazione dei gestori code per la gestione remota”](#) a pagina 188.

## Visualizzazione e nascondimento di gestori code e cluster in IBM MQ Explorer

IBM MQ Explorer può visualizzare più di un gestore code alla volta. Dal pannello Mostra / Nascondi gestore code (selezionabile dal menu per il nodo della struttura ad albero Gestori code), è possibile scegliere se visualizzare le informazioni su un'altra macchina (remota). I gestori code locali vengono rilevati automaticamente.

Per visualizzare un gestore code remoto:

1. Fare clic con il pulsante destro del mouse sul nodo della struttura **Gestori code** , quindi selezionare **Mostra / Nascondi gestori code**.
2. Fare clic su **Aggiungi**. Viene visualizzato il pannello Mostra / nascondi gestori code.
3. Immettere il nome del gestore code remoto e il nome host o l'indirizzo IP nei campi forniti.

Il nome host o l'indirizzo IP viene utilizzato per stabilire una connessione client al gestore code remoto utilizzando il relativo canale di connessione server predefinito, SYSTEM.ADMIN.SVRCONN o un canale di connessione server definito dall'utente.

4. Fare clic su **Fine**.

Il pannello Mostra / nascondi gestori code visualizza anche un elenco di tutti i gestori code visibili. È possibile utilizzare questo pannello per nascondere i gestori code dalla vista di navigazione.

Se il IBM MQ Explorer visualizza un gestore code che è un membro di un cluster, il cluster viene rilevato e visualizzato automaticamente.

Per esportare l'elenco di gestori code remoti da questo pannello:

1. Chiudere il pannello Mostra / nascondi gestori code.
2. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero **IBM MQ** più alto nel riquadro di navigazione di IBM MQ Explorer, quindi selezionare **Esporta impostazioni IBM MQ Explorer**
3. Fare clic su **IBM MQ Explorer > IBM MQ Explorer Impostazioni**
4. Selezionare **Informazioni di connessione > Gestori code remoti**.

5. Selezionare un file in cui memorizzare le impostazioni esportate.
6. Infine, fare clic su **Fine** per esportare le informazioni di connessione del gestore code remoto nel file specificato.

Per importare un elenco di gestori code remoti:

1. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero **IBM MQ** più alto nel riquadro di navigazione di IBM MQ Explorer, quindi selezionare **Importa impostazioni IBM MQ Explorer**
2. Fare clic su **IBM MQ Explorer > IBM MQ Explorer Impostazioni**
3. Fare clic su **Sfoglia** e passare al percorso del file che contiene le informazioni di connessione del gestore code remoto.
4. Fare clic su **Apri**. Se il file contiene un elenco di gestori code remoti, la casella **Informazioni di connessione > Gestori code remoti** è selezionata.
5. Infine, fare clic su **Fine** per importare le informazioni di connessione del gestore code remoto in IBM MQ Explorer.

## Appartenenza al cluster e IBM MQ Explorer

IBM MQ Explorer richiede informazioni sui gestori code che sono membri di un cluster.

Se un gestore code è membro di un cluster, il nodo della struttura ad albero del cluster verrà popolato automaticamente.

Se i gestori code diventano membri dei cluster mentre IBM MQ Explorer è in esecuzione, è necessario mantenere il IBM MQ Explorer con i dati di amministrazione aggiornati sui cluster in modo che possano comunicare in modo efficace con essi e visualizzare le informazioni sul cluster corrette quando richiesto. Per fare ciò, IBM MQ Explorer ha bisogno delle seguenti informazioni:

- Il nome di un gestore code del repository
- Il nome della connessione del gestore code del repository se si trova su un gestore code remoto

Con queste informazioni, IBM MQ Explorer può:

- Utilizzare il gestore code del repository per ottenere un elenco di gestori code nel cluster.
- Gestire i gestori code che sono membri del cluster e si trovano su piattaforme e livelli di comando supportati.

La somministrazione non è possibile se:

- Il repository scelto diventa non disponibile. IBM MQ Explorer non passa automaticamente a un repository alternativo.
- Impossibile contattare il repository scelto tramite TCP/IP.
- Il repository scelto è in esecuzione su un gestore code in esecuzione su una piattaforma e a livello di comando non supportato da IBM MQ Explorer.

I membri del cluster che possono essere gestiti possono essere locali o remoti se possono essere contattati utilizzando TCP/IP. Il IBM MQ Explorer si connette ai gestori code locali che sono membri direttamente di un cluster, senza utilizzare una connessione client.

## Conversione dati per IBM MQ Explorer

IBM MQ Explorer funziona in CCSID 1208 (UTF-8). Ciò consente a IBM MQ Explorer di visualizzare correttamente i dati dai gestori code remoti. Se ci si connette direttamente a un gestore code, o utilizzando un gestore code intermedio, IBM MQ Explorer richiede che tutti i messaggi in entrata vengano convertiti in CCSID 1208 (UTF-8).

Viene emesso un messaggio di errore se si tenta di stabilire una connessione tra IBM MQ Explorer e un gestore code con un CCSID che IBM MQ Explorer non riconosce.

Le conversioni supportate sono descritte in [Conversione codepage](#).

L'applicazione della barra delle applicazioni IBM MQ visualizza un'icona nella barra delle applicazioni Windows sul server. L'icona fornisce lo stato corrente di IBM MQ e un menu da cui è possibile eseguire alcune semplici azioni.

Su Windows, l'icona IBM MQ si trova nella barra delle applicazioni del server e viene sovrapposta con un simbolo di stato codificato a colori, che può avere uno dei seguenti significati:

**Verde**

Funzionamento corretto; nessun avviso al momento

**Blu**

Indeterminato; IBM MQ è in fase di avvio o di arresto

**Giallo**

Avviso; uno o più servizi sono in errore o hanno già avuto esito negativo

Per visualizzare il menù, fare clic con il tasto destro del mouse su IBM MQ . Dal menu è possibile eseguire le seguenti operazioni:

- Fare clic su **Apri** per aprire il Controllo segnalazioni IBM MQ .
- Fare clic su **Esci** per uscire dall'applicazione della barra delle attività IBM MQ .
- Fare clic su **IBM MQ Explorer** per avviare IBM MQ Explorer.
- Fare clic su **Arresta IBM MQ** per arrestare IBM MQ.
- Fare clic su **Informazioni su IBM MQ** per visualizzare le informazioni su IBM MQ Controllo segnalazioni.

## L'applicazione di monitoraggio avvisi IBM MQ (solo Windows)

Il controllo degli avvisi IBM MQ è uno strumento di rilevamento degli errori che identifica e registra i problemi con IBM MQ su una macchina locale.

Il monitoraggio avvisi visualizza le informazioni sullo stato corrente dell'installazione locale di un server IBM MQ . Inoltre, monitora Windows ACPI (Advanced Configuration and Power Interface) e garantisce l'applicazione delle impostazioni ACPI.

Dal monitoraggio avvisi IBM MQ , è possibile:

- Accedi direttamente a IBM MQ Explorer
- Visualizza le informazioni relative a tutti gli avvisi in sospenso
- Arrestare il servizio IBM MQ sulla macchina locale
- Instradare i messaggi di avviso sulla rete a un account utente configurabile o a una workstation o a un server Windows

## Utilizzo degli oggetti IBM MQ locali

---

È possibile gestire oggetti IBM MQ locali per supportare programmi applicativi che utilizzano MQI (Message Queue Interface).

### Informazioni su questa attività

In questo contesto, amministrazione locale significa creare, visualizzare, modificare, copiare ed eliminare oggetti IBM MQ .

In aggiunta agli approcci descritti in questa sezione, è possibile utilizzare IBM MQ Explorer per gestire gli oggetti IBM MQ locali. Per ulteriori informazioni, consultare [“Amministrazione mediante IBM MQ Explorer”](#) a pagina 114.

## Procedura

- Utilizzare le informazioni riportate nei seguenti argomenti per semplificare la gestione degli oggetti IBM MQ locali.
  - [Programmi applicativi che utilizzano MQI](#)
  - [“Gestione mediante comandi MQSC” a pagina 11](#)
  - [“Visualizzazione e modifica degli attributi del gestore code” a pagina 130](#)
  - [“Gestione delle code locali” a pagina 133](#)
  - [“Gestione delle code alias” a pagina 145](#)
  - [“Utilizzo delle code modello” a pagina 147](#)
  - [“Gestione dei servizi” a pagina 175](#)
  - [“Gestione degli oggetti per il trigger” a pagina 182](#)

## Uso dei gestori code

È possibile utilizzare i comandi di controllo per avviare e arrestare un gestore code. È possibile utilizzare i comandi MQSC per visualizzare o modificare gli attributi del gestore code.

### Attività correlate

[Creazione di gestori code su più piattaforme](#)

### **Avvio di un gestore code**

Quando si crea un gestore code, è necessario avviarlo per consentirgli di elaborare comandi o chiamate MQI.

### Informazioni su questa attività

È possibile avviare un gestore code utilizzando il comando **strmqm**. Per una descrizione del comando **strmqm** e delle relative opzioni, vedere [strmqm](#).

  In alternativa, sui sistemi Windows e Linux (x86 e x86-64), è possibile avviare un gestore code utilizzando IBM MQ Explorer.

 Su Windows, è possibile avviare un gestore code automaticamente quando il sistema inizia a utilizzare IBM MQ Explorer. Per ulteriori informazioni, consultare [“Amministrazione mediante IBM MQ Explorer” a pagina 114](#).

## Procedura

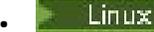
- Per avviare il gestore code utilizzando il comando **strmqm**, immettere il comando seguito dal nome del gestore code che si desidera avviare.

Ad esempio, per avviare un gestore code denominato QMB, immettere il seguente comando:

```
strmqm QMB
```

**Nota:** È necessario utilizzare il comando **strmqm** dall'installazione associata al gestore code che si sta utilizzando. È possibile scoprire a quale installazione è associato un gestore code utilizzando il comando `dspmqr -o installation`.

Il comando **strmqm** non restituisce il comando di controllo fino a quando il gestore code non viene avviato ed è pronto ad accettare le richieste di connessione.

-  

Per avviare un gestore code utilizzando IBM MQ Explorer, completare la seguente procedura:

- a) Apri IBM MQ Explorer.

- b) Nella vista Navigator , selezionare il gestore code.
- c) Fare clic su **Avvia**.

## Risultati

Il gestore code viene avviato.

Se l'avvio del gestore code impiega più di pochi secondi, IBM MQ emette messaggi informativi che descrivono in modo intermittente l'avanzamento dell'avvio.

Multi

## Arresto di un gestore code

È possibile utilizzare il comando **endmqm** per interrompere un gestore code. Questo comando fornisce quattro modi per arrestare un gestore code: un arresto controllato o inattivo, un arresto immediato, un arresto preventivo e un arresto di attesa. In alternativa, su Windows e Linux, è possibile arrestare un gestore code utilizzando IBM MQ Explorer.

## Informazioni su questa attività

Esistono quattro modi per arrestare un singolo gestore code dell'istanza con il comando **endmqm** :

### Chiusura controllata (disattivata)

Per impostazione predefinita, il comando **endmqm** esegue un arresto inattivo del gestore code specificato. Un arresto disattivato attende che tutte le applicazioni connesse si siano disconnesse, pertanto il completamento potrebbe richiedere del tempo.

### arresto immediato

Per un arresto immediato, tutte le chiamate MQI correnti possono essere completate, ma le nuove chiamate non riescono. Questo tipo di arresto non attende la disconnessione delle applicazioni dal gestore code.

### arresto preventivo

Il gestore code viene arrestato immediatamente. Utilizzare questo tipo di arresto solo in circostanze eccezionali, ad esempio quando un gestore code non si arresta come risultato di un normale comando **endmqm** .

### Attendi arresto

Questo tipo di arresto è equivalente a un arresto controllato, ad eccezione del fatto che il controllo viene restituito solo dopo che il gestore code è stato arrestato.

Il comando **endmqm** arresta tutte le istanze di un gestore code a più istanze nello stesso modo in cui arresta un gestore code a istanza singola. È possibile emettere il comando **endmqm** sull'istanza attiva o su una delle istanze in standby di un gestore code a più istanze. Tuttavia, è necessario immettere **endmqm** sull'istanza attiva per terminare il gestore code.

V 9.2.0

Da IBM MQ 9.1.4, è possibile arrestare il gestore code entro un periodo di tempo di destinazione di un numero di secondi specificato; consultare [“Chiusura di un gestore code entro un'ora di destinazione”](#) a pagina 126 per ulteriori dettagli.

Per una descrizione dettagliata del comando **endmqm** e delle sue opzioni, consultare [endmqm](#).

**Suggerimento:** I problemi con la chiusura di un gestore code sono spesso causati dalle applicazioni. Ad esempio, quando le applicazioni:

- Non controllare correttamente i codici di ritorno MQI
- Non richiedere la notifica di un quiesce
- Terminare senza disconnettersi dal gestore code (emettendo una chiamata MQDISC )

Se si verifica un problema quando si tenta di arrestare il gestore code, è possibile interrompere il comando **endmqm** utilizzando Ctrl-C. È quindi possibile immettere un altro comando **endmqm** , ma questa volta con un parametro che specifica il tipo di arresto richiesto.

Come alternativa all'utilizzo del comando **endmqm**, su Windows e Linux, è possibile arrestare un gestore code utilizzando IBM MQ Explorer per eseguire un arresto controllato o immediato.

## Procedura

- Per arrestare il gestore code utilizzando il comando **endmqm**, immettere il comando seguito dal parametro appropriato, se richiesto, e il nome del gestore code che si desidera arrestare.

**Nota:** È necessario utilizzare il comando **endmqm** dall'installazione associata al gestore code che si sta utilizzando. Per individuare a quale installazione è associato un gestore code, utilizzare il comando **dspmqr**:

```
dspmqr -o installation
```

- Per eseguire un arresto controllato (disattivato), immettere il comando **endmqm** come mostrato nel seguente esempio, che arresta un gestore code denominato QMB:

```
endmqm QMB
```

In alternativa, l'immissione del comando **endmqm** con il parametro **-c**, come mostrato nel seguente esempio, equivale a un comando **endmqm QMB**.

```
endmqm -c QMB
```

In entrambi i casi, il controllo viene restituito immediatamente e non si riceve alcuna notifica quando il gestore code è stato arrestato. Se si desidera che il comando attenda che tutte le applicazioni siano state arrestate e che il gestore code sia terminato prima di restituire il controllo all'utente, utilizzare il parametro **-w** come mostrato nel seguente esempio.

```
endmqm -w QMB
```

- Per eseguire un arresto immediato, immettere il comando **endmqm** con il parametro **-i** come mostrato nel seguente esempio:

```
endmqm -i QMB
```

- Per eseguire un arresto preventivo, immettere il comando **endmqm** con il parametro **-p** come mostrato nel seguente esempio:

```
endmqm -p QMB
```



**Attenzione:** Un arresto preventivo può avere conseguenze imprevedibili per le applicazioni connesse. Non utilizzare questa opzione a meno che tutti gli altri tentativi di arrestare il gestore code utilizzando un normale comando **endmqm** non siano riusciti. **ALW** Se l'arresto preventivo non funziona, provare [“Arresto manuale di un gestore code”](#) a pagina 127.

- Per richiedere la [riconnesione client automatica](#), immettere il comando **endmqm** con il parametro **-r**. Questo parametro ha l'effetto di ristabilire la connessione dei client ad altri gestori code nel relativo gruppo di gestori code.

**Nota:** L'arresto di un gestore code utilizzando il comando **endmqm** predefinito non attiva la [riconnesione client automatica](#).

- Per trasferire a un'istanza in standby di un gestore code a più istanze dopo aver arrestato l'istanza attiva, immettere il comando **endmqm** con il parametro **-s** sull'istanza attiva del gestore code a più istanze.

- Per terminare l'istanza in standby di un gestore code a più istanze e lasciare in esecuzione l'istanza attiva, immettere il comando **endmqm** con il parametro **-x** nell'istanza in standby del gestore code a più istanze.



In Windows e Linux, per arrestare il gestore code utilizzando IBM MQ Explorer, completare la seguente procedura:

- Apri IBM MQ Explorer.
- Selezionare il gestore code dalla vista Navigator .
- Fare clic su **Arresta**.  
Viene visualizzato il pannello **Termina gestore code** .
- Selezionare **Controllatao Immediata**.
- Fare clic su **OK**.  
Il gestore code viene arrestato.

### Attività correlate

[Applicazione degli aggiornamenti del livello di manutenzione ai gestori code a più istanze su AIX](#)

[Applicazione degli aggiornamenti del livello di manutenzione ai gestori code a più istanze su Linux](#)

[Applicazione degli aggiornamenti del livello di manutenzione ai gestori code a più istanze su Windows](#)

### Chiusura di un gestore code entro un'ora di destinazione

È possibile terminare il gestore code entro un tempo di destinazione di un numero di secondi specificato, con o senza interrompere le attività essenziali di manutenzione del gestore code.

Esistono due modi per specificare un'ora di destinazione quando si utilizza il comando **endmqm** . L'opzione **-t** consente il completamento delle attività essenziali di manutenzione del gestore code, che potrebbero prolungare la fase di chiusura del gestore code. L'opzione **-tp** interrompe le attività essenziali di manutenzione del gestore code, se necessario, per rispettare il tempo di destinazione specificato.

Quando si specifica un'ora di destinazione, il tipo di arresto **-w**, **-io** **-p** indica il tipo di arresto iniziale.

**Nota:** Un arresto di *immediate* è ancora ordinato, differendo da un arresto di *controlled* principalmente nel modo in cui le applicazioni in esecuzione vengono disattivate. Un arresto di *immediate* continua ad eseguire azioni di manutenzione come la compattazione della coda e la persistenza dei messaggi NPMCLASS (HIGH) che potrebbero richiedere tempo, mentre un arresto limitato nel tempo chiude queste azioni aggiuntive quando interferiscono con il raggiungimento dell'orario di destinazione.

Il gestore code esegue l'escalation del tipo di arresto come necessario, nel tentativo di soddisfare l'ora di destinazione. Ad esempio:

- Una destinazione **-t** di 10 secondi a partire da **-w** potrebbe essere sette secondi di quiesce, due secondi di arresto immediato del gestore code, incluse le operazioni di pulizia, quindi l'arresto immediato senza ulteriori operazioni di pulizia:

```
endmqm -w -t 10 queue_manager
```

- Una destinazione **-tp** di 10 secondi potrebbe essere sette secondi di sospensione, due secondi di arresto immediato del gestore code, inclusa la manutenzione, un secondo di arresto immediato senza ulteriore manutenzione, quindi avviare la chiusura dei processi IBM MQ :

```
endmqm -c -tp 10 queue_manager
```

- Una destinazione **-tp** di due secondi in **-i** potrebbe essere un secondo di arresto immediato del gestore code, inclusa la manutenzione, un secondo di arresto immediato senza ulteriore manutenzione, quindi avviare la chiusura dei processi IBM MQ :

```
endmqm -i -tp 2 queue_manager
```

- Una destinazione di un secondo in **-w** potrebbe essere 0.1 secondi in wait, ad esempio, per un tempo sufficiente a inviare i codici di ritorno IBM MQ alle applicazioni connesse, 0.9 secondi per l'arresto immediato del gestore code, incluso il servizio di pulizia, quindi l'arresto immediato senza ulteriori operazioni di manutenzione, quindi l'avvio della chiusura dei processi IBM MQ .

#### Riferimenti correlati

[endmqm \(fine gestore code\)](#)

### Arresto manuale di un gestore code

Se i metodi standard per l'arresto e la rimozione di un gestore code hanno esito negativo, è possibile provare ad arrestare manualmente il gestore code.

#### Informazioni su questa attività

Il modo standard per arrestare i gestori code consiste nell'utilizzare il comando **endmqm** , come descritto in “Arresto di un gestore code” a pagina 124. Se non è possibile arrestare un gestore code nel modo standard, è possibile provare ad arrestare manualmente un gestore code. Il modo in cui si fa ciò dipende dalla piattaforma che si sta utilizzando.

#### Procedura

-  **Windows**  
Per arrestare un gestore code su Windows, consultare “Arresto manuale di un gestore code su Windows” a pagina 127.
-  **Linux**  **AIX**  
Per arrestare un gestore code su AIX o Linux, fare riferimento a “Arresto manuale di un gestore code su AIX and Linux” a pagina 128.

#### Attività correlate

[Creazione e gestione di gestori code su più piattaforme](#)

#### Riferimenti correlati

[endmqm](#)

### Arresto manuale di un gestore code su Windows

Se non è possibile arrestare un gestore code su Windows utilizzando il comando **endmqm** , è possibile provare ad arrestare manualmente il gestore code arrestando i processi in esecuzione e arrestando il servizio IBM MQ .

#### Informazioni su questa attività

**Suggerimento:** Windows Task Manager e il comando **tasklist** forniscono informazioni limitate sulle attività. Per ulteriori informazioni utili per determinare quali processi sono correlati a uno specifico gestore code, utilizzare uno strumento come *Process Explorer* (*procexp.exe*), disponibile per il download dal sito Web Microsoft all'indirizzo <http://www.microsoft.com>.

Per arrestare un gestore code su Windows, completare la seguente procedura.

#### Procedura

1. Elencare i nomi (ID) dei processi in esecuzione, utilizzando Windows Task Manager.
2. Terminare i processi utilizzando Windows Task Manager o il comando **taskkill** , nel seguente ordine (se sono in esecuzione):

<i>Tabella 7. Processi Windows da arrestare se in esecuzione</i>	
<b>Nome processo</b>	<b>Descrizione</b>
AMQZMUC0	Gestore processi critici
AMQZXMA0	Controllore di Esecuzione
AMQZFUMA	processo OAM
AMQZLAA0	Agent LQM
AMQZLSA0	Agent LQM
AMQZMUFO	Gestore utilità
AMQZMGR0	Controller processo
AMQZMUR0	Gestore processi riavviabile
AMQFQPUB	Processo di pubblicazione - sottoscrizione
AMQFCXBA	Processo di lavoro broker
AMQRMPPA	processo di pooling del processo
AMQCRSTA	Processo di lavoro del responder non sottoposto a thread
AMQCRS6B	Canale ricevente LU62 e connessione client
AMQRRMFA	Il processo del repository (per i cluster)
AMQPCSEA	Il server comandi
RUNMQTRM	Richiama un controllo trigger per un server
RUNMQDLQ	Richiama gestore code di messaggi non recapitabili
RUNMQCHI	Il processo iniziatore del canale
RUNMQLSR	Il processo del listener del canale
VNXMQS	Server di memoria condivisa

- Arrestare il servizio IBM MQ da **Strumenti di gestione > Servizi** nel Windows Pannello di controllo.
- Se sono stati tentati tutti i metodi e il gestore code non è stato arrestato, riavviare il sistema.

### **Arresto manuale di un gestore code su AIX and Linux**

Se non è possibile arrestare un gestore code su AIX o Linux utilizzando il comando **endmqm**, è possibile provare ad arrestare manualmente il gestore code arrestando i processi in esecuzione e arrestando il servizio IBM MQ.

#### **Informazioni su questa attività**

Per arrestare un gestore code su AIX o Linux, completare la seguente procedura.

Se si arresta il gestore code manualmente, FFST potrebbe essere utilizzato e i file FDC posizionati in `/var/mqm/errors`. Non deve essere considerato come un difetto nel gestore code.

Il gestore code verrà riavviato normalmente, anche dopo averlo arrestato utilizzando questo metodo di arresto manuale.

## Procedura

1. Individuare gli ID processo dei programmi del gestore code ancora in esecuzione utilizzando il comando **ps**.

Ad esempio, se il gestore code è denominato QMNAME, utilizzare il seguente comando:

```
ps -ef | grep QMNAME
```

2. Terminare tutti i processi del gestore code ancora in esecuzione utilizzando il comando **kill**, specificando i PID rilevati utilizzando il comando **ps**.

Per terminare un processo, utilizzare **kill -KILL <pid>** o il comando **kill -9 <pid>** equivalente.

Devi lavorare attraverso i PID che vuoi uccidere, uno per uno, emettendo quel comando ogni volta.

**Importante:** Se si utilizza un segnale diverso da **9(SIGKILL)**, il processo probabilmente non si arresterà e si otterranno risultati imprevedibili.

Terminare i processi nel seguente ordine:

Nome processo	Descrizione
amqzmuc0	Gestore processi critici
amqzma0	Controllore di Esecuzione
amqzfuma	processo OAM
amqzlaa0	Agent LQM
amqzlsa0	Agent LQM
amqzmuf0	Gestore utilità
amqzmur0	Gestore processi riavviabile
amqzmgr0	Controller processo
amqfqpub	Processo di pubblicazione - sottoscrizione
amqfcxba	Processo di lavoro broker
amqrmppa	processo di pooling del processo
amqcrsta	Processo di lavoro del responder non sottoposto a thread
amqcrs6b	Canale ricevente LU62 e connessione client
amqrrmfa	Il processo del repository (per i cluster)
amqpcsea	Il server comandi
runmqtrm	Richiama un controllo trigger per un server
runmqdlq	Richiama gestore code di messaggi non recapitabili
runmqchi	Il processo dell'iniziatore di canali
runmqlsr	Il processo del listener del canale

**Nota:** È possibile utilizzare il comando **kill -9** per terminare i processi che non riescono ad arrestarsi.

## Riavvio di un gestore code

È possibile utilizzare il comando **strmqm** per riavviare un gestore code oppure, su sistemi Windows e Linux x86-64 , è possibile riavviare un gestore code da IBM MQ Explorer.

### Informazioni su questa attività

È possibile riavviare un gestore code utilizzando il comando **strmqm** . Per una descrizione del comando **strmqm** e delle relative opzioni, vedere [strmqm](#).

Su sistemi Windows e Linux x86-64 , è possibile riavviare un gestore code utilizzando IBM MQ Explorer nello stesso modo in cui si avvia un gestore code.

### Procedura

- Per riavviare un gestore code utilizzando il comando **strmqm** , immettere il comando seguito dal nome del gestore code che si desidera riavviare.

Ad esempio, per avviare un gestore code denominato `strmqm saturn.queue.manager`, immettere il seguente comando:

```
strmqm saturn.queue.manager
```

- Per avviare un gestore code utilizzando IBM MQ Explorer, completare la seguente procedura:

- Apri IBM MQ Explorer.
- Nella vista Navigator , selezionare il gestore code.
- Fare clic su **Avvia**.

### Risultati

Il gestore code viene riavviato.

Se il riavvio del gestore code impiega più di pochi secondi, IBM MQ emette messaggi informativi che descrivono in modo intermittente l'avanzamento dell'avvio.

## Visualizzazione e modifica degli attributi del gestore code

È possibile utilizzare i comandi MQSC per visualizzare o modificare gli attributi del gestore code.

### Informazioni su questa attività

Si utilizza il comando **DISPLAY QMGR** per visualizzare i parametri del gestore code per un gestore code e il comando **ALTER QMGR** per modificare i parametri del gestore code per il gestore code locale.

### Procedura

- Per visualizzare gli attributi del gestore code specificato nel comando **runmqsc** , utilizzare il comando MQSC **DISPLAY QMGR** :

```
DISPLAY QMGR
```

Il seguente esempio mostra l'output tipico di questo comando:

```
DISPLAY QMGR
 1 : DISPLAY QMGR
AMQ8408: Display Queue Manager details.
QMNAME(QM1)                                ACCTCONO(DISABLED)
```

```

ACCTINT(1800)
ACCTQ(OFF)
ACTVCONO (DISABLED)
ALTDATA(2012-05-27)
AUTHOREV(DISABLED)
CHAD(DISABLED)
CHADEXIT( )
CLWLDATA( )
CLWLLEN(100)
CLWLUSEQ(LOCAL)
CMDLEVEL(800)
CONFIGEV(DISABLED)
CRTIME(16.14.01)
DEFXMITQ( )
DISTL(YES)
IPADDRV(IPV4)
LOGGEREV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWSNT)
PSNPMSG(DISCARD)
PSSYNCP(IFPER)
PSMODE(ENABLED)
REPOS( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRYP( )
SSLFIPS(NO)
MQ\Data\qmgrs\QM1\ssl\key)
SSLKEYC(0)
STATCHL(OFF)
STATMQI(OFF)
STRSTPEV(ENABLED)
TREELIFE(1800)

ACCTMQI(OFF)
ACTIVREC(MSG)
ACTVTRC (OFF)
ALTTIME(16.14.01)
CCSID(850)
CHADEV(DISABLED)
CHLEV(DISABLED)
CLWLEXIT( )
CLWLMRUC(999999999)
CMDEV(DISABLED)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
CRDATE(2011-05-27)
DEADQ( )
DESCR( )
INHIBTEV(DISABLED)
LOCALEV(DISABLED)
MARKINT(5000)
MAXMSGL(4194304)
MAXPRTY(9)
MONACLS(QMGR)
MONQ(OFF)
PERFMEV(DISABLED)
    PSRTYCNT(5)
PSNPRES(NORMAL)
QMID(QM1_2011-05-27_16.14.01)
REMOTEEV(DISABLED)
REPOSNL( )
SCHINIT(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
SSLKEYR(C:\Program Files\IBM\WebSphere

STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SYNCP
TRIGINT(999999999)

```

**Nota:** SYNCP è un attributo del gestore code di sola lettura.

Il parametro **ALL** è il valore predefinito sul comando **DISPLAY QMGR**. Visualizza tutti gli attributi del gestore code. In particolare, l'output indica il nome del gestore code predefinito, il nome della coda di messaggi non recapitabili e il nome della coda comandi.

È possibile confermare l'esistenza di queste code immettendo il seguente comando:

```
DISPLAY QUEUE (SYSTEM.*)
```

Visualizza un elenco di code che corrispondono alla radice SYSTEM.\*. Le parentesi sono obbligatorie.

- Per modificare gli attributi del gestore code specificato sul comando **runmqsc**, utilizzare il comando MQSC **ALTER QMGR**, specificando gli attributi e valori che si desidera modificare.

Ad esempio, utilizzare i comandi riportati di seguito per modificare gli attributi di `jupiter.queue.manager`:

```
runmqsc jupiter.queue.manager
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

Il comando **ALTER QMGR** modifica la coda di messaggi non recapitabili utilizzata e abilita gli eventi di inibizione.

I parametri non specificati nel comando **ALTER QMGR** fanno sì che i valori esistenti per tali parametri non vengano modificati.

### Attività correlate

[Creazione di gestori code su più piattaforme](#)

### Riferimenti correlati

[Attributi per il gestore code](#)

[runmqsc \(esecuzione comandi MQSC\)](#)

## Eliminazione di un gestore code

È possibile eliminare un gestore code utilizzando il comando **dltmqm**. In alternativa, su sistemi Windows e Linux, è possibile utilizzare IBM MQ Explorer per eliminare un gestore code.

### Prima di iniziare



#### Attenzione:

- L'eliminazione di un gestore code è un passo drastico, poiché si eliminano anche tutte le risorse associate al gestore code, incluse tutte le code, i relativi messaggi e tutte le definizioni di oggetti. Se si utilizza il comando **dltmqm**, non viene visualizzata alcuna richiesta che consenta di cambiare idea; quando si preme il tasto Invio, tutte le risorse associate vengono perse.
- **Windows** In Windows, l'eliminazione di un gestore code rimuove anche il gestore code dall'elenco di avvio automatico (descritto in [“Avvio di un gestore code”](#) a pagina 123). Quando il comando è stato completato, viene visualizzato un messaggio IBM MQ queue manager ending; non viene indicato che il gestore code è stato eliminato.
- L'eliminazione di un gestore code del cluster non lo rimuove dal cluster. Per ulteriori informazioni, consultare le note di utilizzo in [dltmqm](#).

### Informazioni su questa attività

È possibile eliminare un gestore code utilizzando il comando **dltmqm**. Per una descrizione del comando **dltmqm** e delle sue opzioni, consultare [dltmqm](#). Assicurarsi che solo gli amministratori attendibili abbiano l'autorità per utilizzare questo comando. (Per informazioni sulla protezione, vedere [Impostazione della protezione su AIX, Linux, and Windows](#).)

In alternativa, su sistemi Windows e Linux (piattaformex86 e x86-64), è possibile eliminare un gestore code utilizzando IBM MQ Explorer.

### Procedura

- Per eliminare un gestore code utilizzando il comando **dltmqm**, completare la seguente procedura:
  - a) Chiudere il gestore code.
  - b) Emetti il seguente comando:

```
dltmqm QMB
```

**Nota:** È necessario utilizzare il comando **dltmqm** dall'installazione associata al gestore code che si sta utilizzando. È possibile scoprire a quale installazione è associato un gestore code utilizzando il comando `dspmq -o installation`.

- Per eliminare un gestore code utilizzando IBM MQ Explorer, completare la seguente procedura:
  - a) Apri IBM MQ Explorer.
  - b) Nella vista Navigator, selezionare il gestore code.
  - c) Se il gestore code non è arrestato, arrestarlo.  
Per arrestare il gestore code, fare clic con il pulsante destro del mouse su di esso e fare clic su **Arresta**.
  - d) Eliminare il gestore code.  
Per eliminare il gestore code, fare clic con il tasto destro del mouse su di esso e fare clic su **Elimina**.

## Risultati

Il gestore code è stato eliminato.

## Arresto dei canali MQI

Quando si immette un comando STOP CHANNEL su un canale di connessione server, è possibile scegliere quale metodo utilizzare per arrestare il canale di connessione client. Ciò significa che un canale client che emette una chiamata di attesa MQGET può essere controllato ed è possibile decidere come e quando arrestare il canale.

Il comando STOP CHANNEL può essere emesso con tre modi, indicando come deve essere arrestato il canale:

### Tempo di sospensione

Arresta il canale dopo che sono stati elaborati tutti i messaggi correnti.

Se la condivisione delle conversazioni è abilitata, IBM MQ MQI client viene a conoscenza della richiesta di arresto in modo tempestivo; questo tempo dipende dalla velocità della rete. L'applicazione client viene a conoscenza della richiesta di arresto come risultato dell'emissione di una chiamata successiva a IBM MQ.

### Forza

Arresta immediatamente il canale.

### Termina

Arresta immediatamente il canale. Se il canale è in esecuzione come un processo, può terminare il processo del canale o se il canale è in esecuzione come un thread, il suo thread.

Questo è un processo a più fasi. Se viene utilizzata la modalità terminate, viene effettuato un tentativo di arresto del canale di connessione server, prima con la modalità quiesce, quindi con la modalità force e, se necessario, con la modalità terminate. Il client può ricevere codici di ritorno differenti durante le diverse fasi di terminazione. Se il processo o il thread viene terminato, il client riceve un errore di comunicazione.

I codici di ritorno restituiti all'applicazione variano in base alla chiamata MQI emessa e al comando STOP CHANNEL emesso. Il client riceverà un codice di ritorno MQRC\_CONNECTION\_QUIESCING o MQRC\_CONNECTION\_BROKEN. Se un client rileva MQRC\_CONNECTION\_QUIESCING, deve tentare di completare la transazione corrente e terminare. Ciò non è possibile con MQRC\_CONNECTION\_BROKEN. Se il client non completa la transazione e termina abbastanza velocemente, otterrà CONNECTION\_BROKEN dopo pochi secondi. Un comando STOP CHANNEL con MODE (FORCE) o MODE (TERMINATE) è più probabile che risulti in un CONNECTION\_BROKEN rispetto a MODE (QUIESCE).

### Concetti correlati

[Canali](#)

## Gestione delle code locali

Questa sezione contiene esempi di alcuni comandi MQSC che è possibile utilizzare per gestire code locali, modello e alias.

Consultare [Comandi MQSC](#) per informazioni dettagliate su questi comandi.

### Riferimenti correlati

[Limitazioni di denominazione per code](#)

[Limitazioni di denominazione per altri oggetti](#)

## Definizione di una coda locale con DEFINE QLOCAL

Per un'applicazione, il gestore code locale è il gestore code a cui è connessa l'applicazione. Le code gestite dal gestore code locale vengono dette locali per tale gestore code.

## Informazioni su questa attività

Utilizzare il comando MQSC **DEFINE QLOCAL** per creare una coda locale. È anche possibile utilizzare il valore predefinito definito nella definizione della coda locale predefinita oppure è possibile modificare le caratteristiche della coda da quelle della coda locale predefinita.

**Nota:** La coda locale predefinita è denominata SYSTEM.DEFAULT.LOCAL.QUEUE e viene creato sull'installazione del sistema.

## Procedura

- Per creare una coda locale, immettere il comando **DEFINE QLOCAL** come mostrato nel seguente esempio.

In questo esempio, il comando **DEFINE QLOCAL** definisce una coda denominata ORANGE.LOCAL.QUEUE con queste caratteristiche:

- È abilitato per le ricezioni, abilitato per gli inserimenti e opera su una base di ordine di priorità.
- Si tratta di una coda *normale* ; non è una coda di iniziazione o di trasmissione e non genera messaggi trigger.
- La lunghezza massima della coda è 5000 messaggi; la lunghezza massima del messaggio è 4194304 byte.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT(ENABLED) +
  GET(ENABLED) +
  NOTRIGGER +
  MSGDLVSQ(PRIORITY) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  USAGE(NORMAL)
```

### Note:

1. Ad eccezione del valore per la descrizione, tutti i valori di attributo mostrati nell'esempio sono i valori predefiniti. Questi esempi sono inclusi a scopo illustrativo. È possibile ometterli se si è certi che i valori predefiniti sono quelli desiderati o non sono stati modificati. Consultare anche [“Visualizzazione degli attributi dell'oggetto predefiniti con DISPLAY QUEUE”](#) a pagina 134.
2. **USAGE (NORMAL)** indica che questa coda non è una coda di trasmissione.
3. Se si dispone già di una coda locale sullo stesso gestore code con il nome ORANGE.LOCAL.QUEUE, questo comando ha esito negativo. Utilizzare l'attributo **REPLACE** se si desidera sovrascrivere la definizione esistente di una coda, ma consultare anche [“Modifica degli attributi della coda locale con ALTER QLOCAL o DEFINE QLOCAL”](#) a pagina 136.

### Riferimenti correlati

[DEFINE QLOCAL](#)

## Visualizzazione degli attributi dell'oggetto predefiniti con DISPLAY QUEUE

È possibile utilizzare il comando **DISPLAY QUEUE** per visualizzare gli attributi presi dall'oggetto predefinito quando è stato definito un oggetto IBM MQ .

## Informazioni su questa attività

Quando si definisce un oggetto IBM MQ , vengono utilizzati tutti gli attributi non specificati dall'oggetto predefinito. Ad esempio, quando si definisce una coda locale, la coda eredita tutti gli attributi omissi nella definizione dalla coda locale predefinita, denominata SYSTEM.DEFAULT.LOCAL.QUEUE. È possibile utilizzare il comando **DISPLAY QUEUE** per visualizzare esattamente questi attributi.

## Procedura

- Per visualizzare gli attributi oggetto predefiniti per una coda locale, utilizzare il seguente comando:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

La sintassi del comando **DISPLAY** è diversa da quella del comando **DEFINE** corrispondente. Sul comando **DISPLAY** è possibile fornire solo il nome della coda, mentre sul comando **DEFINE** è necessario specificare il tipo di coda, ovvero QLOCAL, QALIAS, QMODEL o QREMOTE.

È possibile visualizzare in modo selettivo gli attributi specificandoli singolarmente. Ad esempio:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH;
```

Questo comando visualizza i tre attributi specificati nel modo seguente:

```
AMQ8409: Display Queue details.  
QUEUE (ORANGE.LOCAL.QUEUE) TYPE (QLOCAL)  
CURDEPTH (0) MAXDEPTH (5000)  
MAXMSGL (4194304)
```

**CURDEPTH** è la profondità della coda corrente, ossia il numero di messaggi sulla coda. Questo è un attributo utile da visualizzare, perché monitorando la profondità della coda è possibile assicurarsi che la coda non si riempia.

## Riferimenti correlati

[VISUALIZZAZIONE CODA](#)

[code DEFINE](#)

## Copia di una definizione di coda locale con DEFINE QLOCAL

È possibile copiare una definizione di coda utilizzando l'attributo **LIKE** nel comando **DEFINE QLOCAL**.

## Informazioni su questa attività

È possibile utilizzare il comando **DEFINE** con l'attributo **LIKE** per creare una coda con gli stessi attributi della coda specificata, piuttosto che quelli della coda locale predefinita del sistema. È anche possibile utilizzare questo formato del comando **DEFINE** per copiare una definizione di coda, ma sostituire una o più modifiche agli attributi dell'originale.

### Note:

- Quando si utilizza l'attributo **LIKE** su un comando **DEFINE**, si stanno copiando solo gli attributi della coda. Non si stanno copiando i messaggi sulla coda.
- Se si definisce una coda locale, senza specificare **LIKE**, è uguale a:

```
DEFINE LIKE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

## Procedura

- Per creare una coda con gli stessi attributi della coda specificata, piuttosto che quelli della coda locale predefinita di sistema, immettere il comando **DEFINE** come mostrato nel seguente esempio. Immettere il nome della coda da copiare esattamente come è stata immessa al momento della creazione della coda. Se il nome contiene caratteri minuscoli, racchiuderlo tra virgolette singole.

Questo esempio crea una coda con gli stessi attributi della coda ORANGE.LOCAL.QUEUE, piuttosto che quelli della coda locale predefinita del sistema:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

- Per copiare una definizione di coda, ma sostituire una o più modifiche agli attributi dell'originale, immettere il comando **DEFINE** come mostrato nel seguente esempio.

Questo comando copia gli attributi della coda ORANGE.LOCAL.QUEUE alla coda THIRD.QUEUE, ma specifica che la lunghezza massima del messaggio nella nuova coda deve essere di 1024 byte, invece di 4194304:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

### Riferimenti correlati

[code DEFINE](#)

## Modifica degli attributi della coda locale con ALTER QLOCAL o DEFINE QLOCAL

È possibile modificare gli attributi della coda in due modi, utilizzando il comando **ALTER QLOCAL** o il comando **DEFINE QLOCAL** con l'attributo **REPLACE**.

### Informazioni su questa attività

È possibile utilizzare l'attributo **REPLACE** del comando **ALTER** e **DEFINE** per sostituire una definizione esistente con la nuova definizione specificata. La differenza tra l'utilizzo di **ALTER** e **DEFINE** è che **ALTER** con **REPLACE** non modifica parametri non specificati, ma **DEFINE** con **REPLACE** imposta tutti i parametri.

### Procedura

- Per modificare gli attributi della coda, utilizzare il comando **ALTER** o il comando **DEFINE** come mostrato nei seguenti esempi.

In questi esempi, la lunghezza massima del messaggio sulla coda ORANGE.LOCAL.QUEUE viene ridotto a 10.000 byte.

- Utilizzando il comando **ALTER** :

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Questo comando modifica un singolo attributo, quello della lunghezza massima del messaggio; tutti gli altri attributi rimangono uguali.

- Utilizzando il comando **DEFINE** con l'opzione **REPLACE**, ad esempio:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Questo comando modifica non solo la lunghezza massima del messaggio, ma anche tutti gli altri attributi, ai quali vengono assegnati valori predefiniti. Quindi, ad esempio, se la coda è stata precedentemente inibita, viene modificata in abilitata all'inserimento poiché l'inserimento abilitato è il valore predefinito, come specificato dalla coda SYSTEM.DEFAULT.LOCAL.QUEUE.

Se si riduce la lunghezza massima del messaggio su una coda esistente, i messaggi esistenti non vengono influenzati. Qualsiasi nuovo messaggio, tuttavia, deve soddisfare i nuovi criteri.

### Riferimenti correlati

[Code ALTER](#)

[MODIFICA QLOCAL](#)

[code DEFINE](#)

[DEFINE QLOCAL](#)

## Cancellazione di una coda locale con CLEAR QLOCAL

È possibile utilizzare il comando **CLEAR QLOCAL** per cancellare una coda locale.

### Prima di iniziare

Non è possibile cancellare una coda se:

- Ci sono messaggi senza commit che sono stati inseriti nella coda nel punto di sincronizzazione.
- Un'applicazione ha attualmente la coda aperta.

### Informazioni su questa attività

Se si desidera cancellare una coda locale utilizzando il comando **CLEAR QLOCAL**, il nome della coda deve essere definito sul gestore code locale.

**Nota:** Non vi è alcuna richiesta che consente di cambiare idea; una volta premuto il tasto Invio, i messaggi vengono persi.

### Procedura

Per cancellare i messaggi da una coda locale, utilizzare **CLEAR QLOCAL** come mostrato nel seguente esempio.

In questo esempio, tutti i messaggi vengono eliminati da una coda locale denominata MAGENTA.QUEUE:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

### Riferimenti correlati

[CANCELLA QLOCAL](#)

## Cancellazione di una coda locale con DELETE QLOCAL

È possibile utilizzare il comando MQSC **DELETE QLOCAL** per eliminare una coda locale.

### Informazioni su questa attività

Una coda non può essere eliminata se contiene messaggi di cui non è stato eseguito il commit.

Se una coda ha uno o più messaggi di cui è stato eseguito il commit e nessun messaggio di cui non è stato eseguito il commit, può essere eliminata solo se si specifica l'opzione **PURGE**. L'eliminazione viene quindi eseguita anche se ci sono messaggi di cui è stato eseguito il commit sulla coda denominata e anche questi messaggi vengono eliminati.

Specificando **NOPURGE** invece di **PURGE** si garantisce che la coda non venga eliminata se contiene messaggi di cui è stato eseguito il commit.

### Procedura

- Per eliminare una coda locale, utilizzare il comando **DELETE QLOCAL** come mostrato nel seguente esempio.

Questo esempio elimina la coda PINK.QUEUE se nella coda non sono presenti messaggi di cui è stato eseguito il commit:

```
DELETE QLOCAL (PINK.QUEUE) NOPURGE
```

Questo esempio elimina la coda PINK.QUEUE anche se sulla coda sono presenti messaggi di cui è stato eseguito il commit:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

## Riferimenti correlati

ELIMINA QLOCALE

## Ricerca delle code con il programma di esempio

IBM MQ fornisce un browser della coda di esempio che è possibile utilizzare per esaminare il contenuto dei messaggi su una coda.

## Informazioni su questa attività

Il browser viene fornito nei formati di origine ed eseguibile nelle seguenti ubicazioni, dove `MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ.

**Windows** In Windows, i nomi file e i percorsi per il browser della coda di esempio sono i seguenti:

### Sorgente

```
MQ_INSTALLATION_PATH\tools\c\samples\
```

### Eseguibile

```
MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe
```

**Linux** **AIX** In AIX and Linux, i nomi file e i percorsi sono i seguenti:

### Sorgente

```
MQ_INSTALLATION_PATH/samp/amqsbcg0.c
```

### Eseguibile

```
MQ_INSTALLATION_PATH/samp/bin/amqsbcg
```

## Procedura

- Per eseguire il programma di esempio, immettere un comando come mostrato nel seguente esempio. Il programma di esempio richiede due parametri di immissione, il nome della coda su cui verranno esaminati i messaggi e il gestore code proprietario di tale coda. Ad esempio:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

## Risultati

I tipici risultati di questo comando vengono mostrati nel seguente esempio:

```
AMQSBCG0 - starts here
*****

MQOPEN - 'SYSTEM.ADMIN.QMGR.EVENT'

MQGET of message number 1
****Message descriptor****

StrucId : 'MD ' Version : 2
Report  : 0 MsgType : 8
Expiry  : -1 Feedback : 0
Encoding : 546 CodedCharSetId : 850
Format  : 'MQEVENT '
Priority : 0 Persistence : 0
MsgId   : X'414D512073617475726E2E71756575650005D30033563DB8'
CorrelId : X'0000000000000000000000000000000000000000000000000000000000000000'
BackoutCount : 0
```

```

ReplyToQ      : '          '
ReplyToQMgr   : 'saturn.queue.manager'
** Identity Context
UserIdentifier : '          '
AccountingToken :
  X'0000000000000000000000000000000000000000000000000000000000000000'
ApplIdentityData : '          '
** Origin Context
PutApplType   : '7'
PutApplName   : 'saturn.queue.manager'
PutDate       : '19970417'   PutTime : '15115208'
ApplOriginData : '          '

GroupId : X'0000000000000000000000000000000000000000000000000000000000000000'
MsgSeqNumber : '1'
Offset       : '0'
MsgFlags     : '0'
OriginalLength : '104'

**** Message ****

length - 104 bytes

00000000: 0700 0000 2400 0000 0100 0000 2C00 0000 | .....→.....|
00000010: 0100 0000 0100 0000 0100 0000 AE08 0000 | .....|
00000020: 0100 0000 0400 0000 4400 0000 DF07 0000 | .....D.....|
00000030: 0000 0000 3000 0000 7361 7475 726E 2E71 | ...0...saturn.q|
00000040: 7565 7565 2E6D 616E 6167 6572 2020 2020 | ueue.manager   |
00000050: 2020 2020 2020 2020 2020 2020 2020 2020 |                |
00000060: 2020 2020 2020 2020

No more messages
MQCLOSE
MQDISC

```

**Riferimenti correlati**

[Il programma di esempio Browser](#)

**Abilitazione di code grandi**

IBM MQ supporta code più grandi di 2 TB.

**Windows** Sui sistemi Windows , il supporto per file di grandi dimensioni è disponibile senza alcuna ulteriore abilitazione.

**Linux AIX** Su sistemi AIX and Linux , è necessario abilitare esplicitamente il supporto per file di grandi dimensioni prima di poter creare file di coda di più gigabyte o terabyte. Per informazioni su come eseguire questa operazione, consultare la documentazione del sistema operativo.

Alcuni programmi di utilità, come tar, non sono in grado di gestire file di più gigabyte o terabyte. Prima di attivare il supporto file di grandi dimensioni, consultare la documentazione del sistema operativo per informazioni sulle restrizioni relative ai programmi di utilità utilizzati.

Per informazioni sulla pianificazione della quantità di memoria necessaria per le code, consultare [MQ Performance documents](#) per report sulle prestazioni specifici della piattaforma.

**V 9.2.0** Da IBM MQ 9.1.5, è possibile controllare la dimensione dei file della coda utilizzando un nuovo attributo sulle code locali e modello. Per ulteriori informazioni, consultare [“Modifica dei file di coda IBM MQ”](#) a pagina 139.

**V 9.2.0 Multi Modifica dei file di coda IBM MQ**

Da IBM MQ 9.2.0, è possibile controllare la dimensione dei file di coda utilizzando un attributo sulle code locali e modello. È possibile visualizzare la dimensione corrente di un file di coda e la dimensione massima che è attualmente in grado di raggiungere (in base alla dimensione del blocco attualmente in uso in tale file), utilizzando due attributi di stato della coda.

## Attributo utilizzato per modificare i file della coda

L'attributo sulle code locali e modello è:

### MAXFSIZE

Indica la dimensione massima del file di coda utilizzato dalla coda, in megabyte.

Consultare [MAXFSIZE](#) e “[Modifica della dimensione di un file di coda IBM MQ](#)” a pagina 140 per ulteriori informazioni.

L'attributo PCF per questo attributo è MQIA\_MAX\_Q\_FILE\_SIZE; consultare [Modifica, copia e crea coda](#).

I due attributi sullo stato della coda sono:

### CURFSIZE

Visualizza la dimensione corrente del file di coda in megabyte, arrotondata al megabyte più vicino.

Per ulteriori informazioni, consultare [CURFSIZE](#).

L'attributo PCF per questo attributo è MQIA\_CUR\_Q\_FILE\_SIZE.

### CURMAXFS

Indica la dimensione massima corrente che il file della coda può raggiungere, arrotondata al megabyte più vicino, data la dimensione del blocco corrente in uso su una coda.

Per ulteriori informazioni, consultare [CURMAXFS](#).

L'attributo PCF per questo attributo è MQIA\_CUR\_MAX\_FILE\_SIZE.

Consultare [Coda di interrogazione](#) e [Coda di interrogazione \(risposta\)](#) per ulteriori informazioni su questi due attributi PCF.

È possibile impostare e visualizzare questi attributi utilizzando comandi MQSC, IBM MQ Explorer o amministrative REST API.

**Nota:** È possibile visualizzare solo il valore di MAXFSIZE in IBM MQ Console; non è possibile configurare il valore.

## Dimensione e granularità del blocco

I file di coda sono divisi in segmenti denominati blocchi. Per aumentare la dimensione massima di un file di coda, potrebbe essere necessario che il gestore code modificasse la dimensione del blocco o la granularità della coda.

Se una coda appena definita viene creata con un valore MAXFSIZE elevato, la coda viene creata con una dimensione di blocco adatta. Tuttavia, se il valore MAXFSIZE di una coda esistente è stato aumentato, ad esempio utilizzando il comando ALTER QLOCAL, potrebbe essere necessario consentire lo svuotamento della coda per consentire al gestore code di riconfigurare la coda.

Per ulteriori informazioni, consultare “[Calcolo della quantità di dati che un file di coda IBM MQ può memorizzare](#)” a pagina 142.



**Attenzione:** Alcuni file system e sistemi operativi hanno dei limiti sulla dimensione dell'intero file system o sulla dimensione di un singolo file. È necessario verificare i limiti sui sistemi utilizzati dall'azienda.

### Riferimenti correlati

[MODIFICA CODE](#)

[VISUALIZZAZIONE CODA](#)

[VISUALIZZAZIONE QSTATUS](#)

**V 9.2.0** **Multi** **Modifica della dimensione di un file di coda IBM MQ**

È possibile aumentare o diminuire la dimensione massima di un file di coda.

## Prima di iniziare

Prima di impostare una nuova dimensione per un file di coda, utilizzare il comando `DISPLAY QLOCAL` per visualizzare la dimensione del file di coda che si desidera modificare. Ad esempio, emettere il seguente comando:

```
DISPLAY QLOCAL (SYSTEM.DEFAULT.LOCAL.QUEUE) MAXFSIZE
```

Si riceve il seguente output:

```
AMQ8409I: Display queue details
          QUEUE(SYSTEM.DEFAULT.LOCAL.QUEUE)      TYPE(QLOCAL)
          MAXFSIZE(DEFAULT)
```

che mostra che il valore predefinito della dimensione massima del file di coda è 2,088,960 MB.

## Informazioni su questa attività

Le seguenti procedure mostrano come:

- Ridurre la dimensione massima di un file di coda.
- Aumentare la dimensione massima di un file di coda.



**Attenzione:** È necessario essere prudenti nell'aumentare la dimensione dei file della coda senza considerare il modo in cui vengono scritte le applicazioni e il possibile impatto sulle prestazioni. L'accesso ai messaggi in modo casuale in un file di coda molto grande può essere molto lento.

Se si sta considerando di aumentare la dimensione massima di un file di coda oltre il valore predefinito, è necessario essere prudenti nell'utilizzare i selettori di messaggi come gli ID di correlazione e le stringhe del selettore IBM MQ classes for JMS . I file di coda più grandi sono più adatti per il primo accesso alla coda.

La presenza di grandi quantità di dati in singoli file di coda deve essere eseguita solo su gestori code configurati per la registrazione circolare o in cui l'imaging del supporto non è stato abilitato per la singola coda.

Non è necessario limitare la dimensione delle code SYSTEM poiché ciò potrebbe influire sul funzionamento del gestore code.

## Procedura

### 1. Ridurre la dimensione massima del file di coda

- a) Immettere il seguente comando per creare un file locale denominato SMALLQUEUE, con una dimensione di 500 gigabyte:

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

e si riceve il seguente messaggio: AMQ800:

**Nota:** Se si configura una coda con un valore inferiore alla quantità di dati già presente nel file, i nuovi messaggi non possono essere inseriti nella coda.

Se un'applicazione tenta di inserire un messaggio in un file di coda che non dispone di spazio sufficiente, l'applicazione riceve il codice di ritorno MQRC\_Q\_SPACE\_NOT\_AVAILABLE. Quando un numero sufficiente di messaggi viene letto in modo distruttivo dalla coda, le applicazioni possono iniziare a inserire nuovi messaggi nella coda.

### 2. Aumentare la dimensione massima del file della coda.

- a) Immettere il seguente comando per creare un file locale denominato LARGEQUEUE, con una dimensione di 5 terabyte:

```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```

V 9.2.0

Multi

## Calcolo della quantità di dati che un file di coda IBM MQ può memorizzare

La quantità di dati che è possibile memorizzare su una coda è limitata dalla dimensione dei singoli blocchi in cui è divisa la coda.

### Dimensione e granularità del blocco

La dimensione blocco predefinita è 512 byte. Per supportare file di coda superiori a due terabyte, il gestore code dovrà aumentare la dimensione del blocco.

La dimensione del blocco viene calcolata automaticamente quando si configura MAXFSIZE per una coda, ma la dimensione del blocco modificata non può essere applicata alla coda se la coda contiene già messaggi. Una volta che una coda è vuota, il Gestore code modifica automaticamente la dimensione del blocco per supportare MAXFSIZE configurato.

Il comando DISPLAY QSTATUS ha un nuovo attributo, CURMAXFS, che consente di confermare che una coda è stata modificata per utilizzare una nuova dimensione blocco.

Nel seguente esempio, il valore CURMAXFS di 4177920 conferma che il file della coda è attualmente in grado di aumentare di circa quattro terabyte. Se il valore di MAXFSIZE configurato sulla coda è maggiore del valore di CURMAXFS, il gestore code è ancora in attesa che la coda venga svuotata prima di riconfigurare la dimensione del blocco del file della coda.

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
        QUEUE(LARGEQUEUE)                TYPE(Queue)
        CURMAXFS(4177920)                 CURDEPTH(100000)
```

### Controllo della dimensione di un file di coda

È possibile visualizzare la dimensione corrente di un file di coda su disco, in megabyte, utilizzando l'attributo CURFSIZE nel comando DISPLAY QSTATUS. Ciò può essere utile su piattaforme come IBM MQ Appliance, dove non è possibile accedere direttamente al file system.

```
DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
1 : DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
AMQ8450I: Display queue status details
        QUEUE(SMALLQUEUE)                TYPE(Queue)
        CURDEPTH(4024)                   CURFSIZE(10)
```

**Nota:** Quando una coda ha dei messaggi rimossi, l'attributo CURFSIZE potrebbe non diminuire immediatamente.

In genere, lo spazio inutilizzato in un file di coda viene rilasciato solo nelle seguenti circostanze:

- Quando nessuna applicazione ha la coda aperta
- Dopo 1000 scritture nel log del gestore code o
- Alla chiusura del gestore code

### Riferimenti correlati

[MODIFICA CODE](#)

[VISUALIZZAZIONE QSTATUS](#)

## Gestione delle code remote

Una coda remota è una definizione locale di una coda remota. Vale a dire, una definizione su un gestore code locale che fa riferimento a una coda su un gestore code remoto.

Non è necessario definire una coda remota da una posizione locale, ma il vantaggio è che le applicazioni possono fare riferimento alla coda remota tramite il nome definito localmente invece di dover specificare un nome qualificato dall'ID del gestore code su cui si trova la coda remota.

### Modalità di funzionamento delle definizioni locali delle code remote

Un'applicazione si connette a un gestore code locale ed emette una chiamata MQOPEN . Nella chiamata di apertura, il nome coda specificato è quello di una definizione di coda remota sul gestore code locale. La definizione della coda remota fornisce i nomi della coda di destinazione, del gestore code di destinazione e, facoltativamente, di una coda di trasmissione. Per inserire un messaggio sulla coda remota, l'applicazione emette una chiamata MQPUT , specificando l'handle restituito dalla chiamata MQOPEN . Il gestore code utilizza il nome coda remota e il nome gestore code remoto in un'intestazione di trasmissione all'inizio del messaggio. Queste informazioni vengono utilizzate per instradare il messaggio alla destinazione corretta nella rete.

Come amministratore, è possibile controllare la destinazione del messaggio modificando la definizione della coda remota.

Il seguente esempio mostra il modo in cui un'applicazione inserisce un messaggio in una coda di proprietà di un gestore code remoto. L'applicazione si connette a un gestore code, ad esempio `saturn.queue.manager`. La coda di destinazione è di proprietà di un altro gestore code.

Nella chiamata MQOPEN , l'applicazione specifica questi campi:

Valore campo	Descrizione
<i>ObjectName</i> CYAN.REMOTE.CODA	Specifica il nome locale dell'oggetto coda remota. Definisce la coda di destinazione e il gestore code di destinazione.
<i>ObjectType</i> (coda)	Identifica questo oggetto come coda.
<i>ObjectQmgrName</i> Vuoto o <code>saturn.queue.manager</code>	Questo campo è facoltativo. Se vuoto, viene utilizzato il nome del gestore code locale. (Questo è il gestore code su cui esiste la definizione della coda remota).

In seguito, l'applicazione emette una chiamata MQPUT per inserire un messaggio in questa coda.

Sul gestore code locale, è possibile creare una definizione locale di una coda remota utilizzando i seguenti comandi MQSC:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
RQMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

dove:

#### **QREMOTE (CYAN.REMOTE.QUEUE)**

Specifica il nome locale dell'oggetto coda remota. Questo è il nome che le applicazioni connesse a questo gestore code devono specificare nella chiamata MQOPEN per aprire la coda `AUTOMOBILE.INSURANCE.QUOTE.QUEUE` sul gestore code remoto `jupiter.queue.manager`.

#### **DESCR ('Queue for auto insurance requests from the branches')**

Fornisce ulteriore testo che descrive l'utilizzo della coda.

### **RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)**

Specifica il nome della coda di destinazione sul gestore code remoto. Questa è la coda di destinazione reale per i messaggi inviati dalle applicazioni che specificano il nome della coda CYAN.REMOTE.QUEUE. La coda AUTOMOBILE.INSURANCE.QUOTE.QUEUE deve essere definita come coda locale sul gestore code remoto.

### **RQMNAME (jupiter.queue.manager)**

Specifica il nome del gestore code remoto proprietario della coda di destinazione AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

### **XMITQ (INQUOTE.XMIT.QUEUE)**

Specifica il nome della coda di trasmissione. Questa opzione è facoltativa; se il nome di una coda di trasmissione non viene specificato, viene utilizzata una coda con lo stesso nome del gestore code remoto.

In entrambi i casi, la coda di trasmissione appropriata deve essere definita come una coda locale con un attributo **Usage** che specifica che è una coda di trasmissione (USAGE (XMITQ) nei comandi MQSC).

## **Un modo alternativo di inserire messaggi in una coda remota**

L'uso di una definizione locale di una coda remota non è l'unico modo per inserire messaggi in una coda remota. Le applicazioni possono specificare il nome completo della coda, incluso il nome gestore code remoto, come parte della chiamata MQOPEN. In questo caso, non è necessaria una definizione locale di una coda remota. Tuttavia, ciò significa che le applicazioni devono conoscere o avere accesso al nome del gestore code remoto in fase di runtime.

## **Utilizzo di altri comandi con code remote**

È possibile utilizzare i comandi MQSC per visualizzare o modificare gli attributi di un oggetto coda remota oppure è possibile eliminare l'oggetto coda remota. Ad esempio:

- Per visualizzare gli attributi della coda remota:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Per modificare la coda remota per abilitare gli inserimenti. Ciò non influisce sulla coda di destinazione, ma solo sulle applicazioni che specificano questa coda remota:

```
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)
```

- Per eliminare questa coda remota. Ciò non influisce sulla coda di destinazione, ma solo sulla sua definizione locale:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

**Nota:** Quando si elimina una coda remota, si elimina solo la rappresentazione locale della coda remota. Non si elimina la coda remota stessa o alcun messaggio su di essa.

## **Utilizzo delle definizioni di coda remota come alias**

Oltre a localizzare una coda su un altro gestore code, è anche possibile utilizzare una definizione locale di una coda remota per gli alias del gestore code e gli alias della coda di risposta. Entrambi i tipi di alias vengono risolti tramite la definizione locale di una coda remota. È necessario impostare i canali appropriati affinché il messaggio arrivi a destinazione.

## **Alias del gestore code**

Un alias è il processo con cui il nome del gestore code di destinazione, come specificato in un messaggio, viene modificato da un gestore code sull'instradamento del messaggio. Gli alias dei gestori code sono

importanti perché è possibile utilizzarle per controllare la destinazione dei messaggi all'interno di una rete di gestori code.

A tale scopo, modificare la definizione della coda remota sul gestore code nel punto di controllo. L'applicazione mittente non è a conoscenza del fatto che il nome del gestore code specificato è un alias.

Per ulteriori informazioni sugli alias del gestore code, consultare [Cosa sono gli alias?](#).

## Alias coda di risposta

Facoltativamente, un'applicazione può specificare il nome di una coda di risposta quando inserisce un *messaggio di richiesta* in una coda.

Se l'applicazione che elabora il messaggio estrae il nome della coda di risposta, sa dove inviare il *messaggio di risposta*, se necessario.

Un alias della coda di risposta è il processo mediante il quale una coda di risposta, come specificato in un messaggio di richiesta, viene modificata da un gestore code sull'instradamento del messaggio. L'applicazione mittente non riconosce che il nome della coda di risposta specificato è un alias.

Un alias della coda di risposta consente di modificare il nome della coda di risposta e, facoltativamente, il relativo gestore code. Ciò a sua volta consente di controllare quale instradamento viene utilizzato per i messaggi di risposta.

Per ulteriori informazioni sui messaggi di richiesta, sui messaggi di risposta e sulle code di risposta, consultare [Tipi di messaggio](#) e [Coda di risposta e gestore code](#).

Per ulteriori informazioni sugli alias della coda reply - to, consultare [Cluster e alias della coda reply - to](#).

## Gestione delle code alias

È possibile definire una coda alias per fare riferimento indirettamente ad un'altra coda o argomento.



**Attenzione:** Gli elenchi di distribuzione non supportano l'utilizzo di code alias che puntano agli oggetti argomento. Se una coda alias punta a un oggetto argomento in un elenco di distribuzione, IBM MQ restituisce MQRC\_ALIAS\_BASE\_Q\_TYPE\_ERROR.

La coda a cui fa riferimento una coda alias può essere una delle seguenti:

- Una coda locale (consultare [“Definizione di una coda locale con DEFINE QLOCAL”](#) a pagina 133 ).
- Una definizione locale di una coda remota (consultare [“Gestione delle code remote”](#) a pagina 143 ).
- Un argomento.

Una coda alias non è una coda reale, ma una definizione che si risolve in una coda reale (o di destinazione) al runtime. La definizione della coda alias specifica la coda di destinazione. Quando un'applicazione effettua una chiamata MQOPEN a una coda alias, il gestore code risolve l'alias nel nome coda di destinazione.

Una coda alias non può essere risolta in un'altra coda alias definita localmente. Tuttavia, una coda alias può risolversi in code alias definite altrove nei cluster di cui è membro il gestore code locale. Per ulteriori informazioni, consultare [Risoluzione dei nomi](#) .

Le code alias sono utili per:

- Fornire alle diverse applicazioni diversi livelli di autorizzazioni di accesso alla coda di destinazione.
- Consentire a diverse applicazioni di gestire la stessa coda in modi diversi. (È possibile che si desideri assegnare priorità predefinite differenti o valori di persistenza predefiniti differenti.)
- Semplificazione della manutenzione, della migrazione e del bilanciamento del carico di lavoro. (Forse si desidera modificare il nome della coda di destinazione senza dover modificare l'applicazione, che continua ad utilizzare l'alias.)

Ad esempio, si supponga che un'applicazione sia stata sviluppata per inserire i messaggi su una coda denominata MY.ALIAS.QUEUE. Specifica il nome di questa coda quando effettua una richiesta MQOPEN e, indirettamente, se inserisce un messaggio su questa coda. L'applicazione non è consapevole che la coda è

una coda alias. Per ogni chiamata MQI che utilizza questo alias, il gestore code risolve il nome della coda reale, che può essere una coda locale o una coda remota definita in questo gestore code.

Modificando il valore dell'attributo TARGET, è possibile reindirizzare chiamate MQI a un'altra coda, possibilmente su un altro gestore code. È utile per la manutenzione, la migrazione e il bilanciamento del carico.

## Definizione di una coda alias

Il seguente comando crea una coda alias:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Questo comando reindirizza le chiamate MQI che specificano MY.ALIAS.QUEUE alla coda YELLOW.QUEUE. Il comando non crea la coda di destinazione; le chiamate MQI hanno esito negativo se la coda è YELLOW.QUEUE non esiste in fase di runtime.

Se si modifica la definizione di alias, è possibile reindirizzare le chiamate MQI a un'altra coda. Ad esempio:

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

Questo comando reindirizza le chiamate MQI a un'altra coda, MAGENTA.QUEUE.

È inoltre possibile utilizzare le code alias per fare in modo che una singola coda (la coda di destinazione) sembri avere attributi differenti per applicazioni differenti. Questa operazione viene eseguita definendo due alias, uno per ogni applicazione. Si supponga che vi siano due applicazioni:

- L'applicazione ALPHA può inserire messaggi su YELLOW.QUEUE, ma non è consentito richiamare messaggi da esso.
- L'applicazione BETA può ricevere messaggi da YELLOW.QUEUE, ma non è consentito inserire messaggi su di esso.

Il seguente comando definisce un alias che viene inserito abilitato e viene disabilitato per l'applicazione ALPHA:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (ENABLED) +  
GET (DISABLED)
```

Il seguente comando definisce un alias che viene inserito disabilitato e abilitato per l'applicazione BETA:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (DISABLED) +  
GET (ENABLED)
```

ALPHA utilizza il nome coda ALPHAS.ALIAS.QUEUE nelle sue chiamate MQI; BETA utilizza il nome della coda BETAS.ALIAS.QUEUE. Entrambi accedono alla stessa coda, ma in modi diversi.

È possibile utilizzare gli attributi LIKE e REPLACE quando si definiscono gli alias della coda, nello stesso modo in cui si utilizzano questi attributi con code locali.

## Utilizzo di altri comandi con code alias

È possibile utilizzare i comandi MQSC appropriati per la visualizzazione o la modifica degli attributi della coda alias o per eliminare l'oggetto della coda alias. Ad esempio:

Utilizzare il comando **DISPLAY QALIAS** per visualizzare gli attributi della coda alias:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Utilizzare il comando **ALTER QALIAS** per modificare il nome della coda di base, in cui l'alias si risolve, in cui l'opzione *force* forza la modifica anche se la coda è aperta:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

Utilizzare il comando **DELETE QALIAS** per eliminare questo alias della coda:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Non è possibile eliminare una coda alias se un'applicazione ha attualmente la coda aperta.

### Riferimenti correlati

[MODIFICA QALIAS](#)

[DEFINE QALIAS](#)

[DELETE QALIAS](#)

[Liste di distribuzione](#)

## Utilizzo delle code modello

Un gestore code crea una *coda dinamica* se riceve una chiamata MQI da un'applicazione che specifica un nome coda definito come coda modello. Il nome della nuova coda dinamica viene generato dal gestore code quando viene creata la coda. Una *coda modello* è un template che specifica gli attributi di tutte le code dinamiche da essa create. Le code modello forniscono un metodo conveniente per le applicazioni per creare le code come richiesto.

### Definizione di una coda modello

Utilizzare il comando **DEFINE QMODEL** per definire una coda modello con una serie di attributi nello stesso modo in cui si definisce una coda locale. Le code modello e le code locali hanno la stessa serie di attributi, tranne per il fatto che sulle code modello è possibile specificare se le code dinamiche create sono temporanee o permanenti. (Le code permanenti vengono gestite durante i riavvii del gestore code, mentre quelle temporanee non lo sono.) Ad esempio:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +  
DESCR('Queue for messages from application X') +  
PUT (DISABLED) +  
GET (ENABLED) +  
NOTRIGGER +  
MSGDLVSQ (FIFO) +  
MAXDEPTH (1000) +  
MAXMSGL (2000) +  
USAGE (NORMAL) +  
DEFTYPE (PERMDYN)
```

Questo comando crea una definizione di coda modello. Dall'attributo **DEFTYPE**, è possibile vedere che le code effettive create da questo modello sono code dinamiche permanenti. Gli attributi non specificati vengono automaticamente copiati da `SYSTEM.DEFAULT.MODEL.QUEUE` predefinita.

È possibile utilizzare gli attributi **LIKE** e **REPLACE** quando si definiscono le code modello, nello stesso modo in cui si utilizzano le code locali.

### Utilizzo di altri comandi con code modello

È possibile utilizzare i comandi MQSC appropriati per visualizzare o modificare gli attributi di una coda modello o per eliminare l'oggetto coda modello. Ad esempio:

Utilizzare il comando **DISPLAY QUEUE** per visualizzare gli attributi della coda modello:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Utilizzare il comando **ALTER QMODEL** per modificare il modello per abilitare gli inserimenti su qualsiasi coda dinamica creata da questo modello:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Utilizzare il comando **DELETE QMODEL** per eliminare questa coda modello:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

### Riferimenti correlati

[MODIFICA QMODEL](#)

[DEFINE QMODEL](#)

[DELETE QMODEL](#)

[VISUALIZZAZIONE CODA](#)

## Gestione delle code di messaggi non recapitabili

Ogni gestore code ha generalmente una coda locale da utilizzare come coda di messaggi non recapitabili, in modo che i messaggi che non possono essere consegnati alla loro corretta destinazione possano essere memorizzati per un successivo recupero. Si indica al gestore code la coda dei messaggi non instradabili e si specifica il modo in cui devono essere elaborati i messaggi rilevati su una coda di messaggi non instradabili. L'utilizzo di code di messaggi non recapitabili può influire sulla sequenza di consegna dei messaggi, pertanto è possibile scegliere di non utilizzarle.

Per informare il gestore code della coda di messaggi non recapitabili, specificare un nome coda di messaggi non recapitabili nel comando **crtmqm** (`crtmqm -u DEAD.LETTER.QUEUE`, ad esempio) o utilizzando l'attributo **DEADQ** nel comando **ALTER QMGR** per specificarne uno in un secondo momento. È necessario definire la coda di messaggi non recapitabili prima di utilizzarla.

Una coda di messaggi non recapitabili di esempio denominata SYSTEM.DEAD.LETTER.QUEUE è disponibile con il prodotto. Questa coda viene creata automaticamente quando si crea il gestore code. Se necessario, è possibile modificare questa definizione e ridenominarla.

Una coda di messaggi non recapitabili non ha requisiti speciali tranne che:

- Deve essere una coda locale
- Il suo attributo MAXMSGL (lunghezza massima del messaggio) deve abilitare la coda a contenere i messaggi più grandi che il gestore code deve gestire **più** la dimensione dell'intestazione dei messaggi non instradabili (MQDLH)

L'utilizzo di code di messaggi non recapitabili può influire sulla sequenza di consegna dei messaggi, pertanto è possibile scegliere di non utilizzarle. Impostare l'attributo del canale USEDQ per determinare se la coda di messaggi non recapitabili viene utilizzata quando i messaggi non possono essere consegnati. Questo attributo può essere configurato in modo che alcune funzioni del gestore code utilizzino la coda di messaggi non recapitabili, mentre altre non lo utilizzano. Per ulteriori informazioni sull'utilizzo dell'attributo del canale USEDQ su diversi comandi MQSC, consultare [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) e [DISPLAY CLUSQMGR](#).

IBM MQ fornisce un gestore code di messaggi non recapitabili che consente di specificare il modo in cui i messaggi trovati in una coda di messaggi non recapitabili devono essere elaborati o rimossi. Consultare [“Elaborazione di messaggi su una coda di messaggi non instradabili IBM MQ”](#) a pagina 149.

### Concetti correlati

[Code di messaggi non recapitabili](#)

## Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

## Riferimenti correlati

[Gestore code ALTER](#)

[crtmqm \(crea gestore code\)](#)

## Elaborazione di messaggi su una coda di messaggi non instradabili IBM MQ

Per elaborare i messaggi su una DLQ (dead - letter queue), IBM MQ fornisce un gestore DLQ predefinito. Il gestore mette in corrispondenza i messaggi sulla DLQ con voci in una tabella di regole definita dall'utente.

I messaggi possono essere inseriti su una DLQ da gestori code, MCA (Message Channel Agent) e applicazioni. Tutti i messaggi sulla DLQ devono avere come prefisso una struttura *dead-letter header*, MQDLH. I messaggi inseriti nella DLQ da un gestore code o da un agente del canale dei messaggi hanno sempre questa intestazione; le applicazioni che inserendo i messaggi nella DLQ devono fornire questa intestazione. Il campo *Motivo* della struttura MQDLH contiene un codice motivo che identifica il motivo per cui il messaggio si trova nella DLQ.

Tutti gli ambienti IBM MQ richiedono una routine per elaborare regolarmente i messaggi sulla DLQ. IBM MQ fornisce una routine predefinita, denominata *gestore code di messaggi non instradabili* (il gestore DLQ), richiamata utilizzando il comando `runmqdlq`.

Le istruzioni per l'elaborazione dei messaggi sul DLQ vengono fornite al gestore DLQ mediante una *tabella di regole* scritta dall'utente. Ciò significa che il gestore DLQ mette in corrispondenza i messaggi sulla DLQ con le voci nella tabella delle regole; quando un messaggio DLQ corrisponde a una voce della tabella delle regole, il gestore DLQ esegue l'azione associata a tale voce.

## Concetti correlati

[Code di messaggi non recapitabili](#)

## Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

## **Il gestore code di messaggi non instradabili su IBM i.**

Cos'è il gestore code di messaggi non recapitabili IBM i e come viene richiamato?

Una *coda di messaggi non recapitabili* (DLQ), a volte indicata come *coda di messaggi non recapitati*, è una coda di congelamento per i messaggi che non possono essere consegnati alle relative code di destinazione. Ogni gestore code in una rete deve avere una DLQ associata.

**Nota:** È spesso preferibile evitare di inserire messaggi in una DLQ. Per informazioni sull'utilizzo e la prevenzione dei DLQ, consultare [“Gestione delle code di messaggi non recapitabili”](#) a pagina 148.

I gestori code, gli agenti del canale dei messaggi e le applicazioni possono inserire messaggi nella DLQ. Tutti i messaggi sulla DLQ devono avere come prefisso una struttura *dead-letter header*, MQDLH. I messaggi immessi sulla DLQ da un gestore code o da un agente del canale dei messaggi hanno sempre un MQDLH. Fornire sempre un MQDLH alle applicazioni che inserendo i messaggi sul DLQ. Il campo *Motivo* della struttura MQDLH contiene un codice motivo che identifica il motivo per cui il messaggio si trova nella DLQ.

In tutti gli ambienti IBM MQ, deve essere presente una routine che viene eseguita regolarmente per elaborare i messaggi sulla DLQ. IBM MQ fornisce una routine predefinita, denominata *gestore code di messaggi non instradabili* (il programma di gestione DLQ), richiamata utilizzando il comando `STRMQMDLQ`. Una *tabella regole* scritta dall'utente fornisce istruzioni al gestore DLQ, per l'elaborazione dei messaggi sulla DLQ. Ossia, il gestore DLQ mette in corrispondenza i messaggi sulla DLQ con le voci nella tabella delle regole. Quando un messaggio DLQ corrisponde ad una voce nella tabella delle regole, il gestore DLQ esegue l'azione associata a tale voce.

## Richiamo del gestore DLQ

Utilizzare il comando STRMQMDLQ per richiamare il programma di gestione DLQ. È possibile denominare la DLQ che si desidera elaborare e il gestore code che si desidera utilizzare in due modi:

- Come parametri per STRMQMDLQ dalla richiesta comandi. Ad esempio:

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- Nella tabella delle regole. Ad esempio:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE)
```

**Nota:** La tabella delle regole è un membro all'interno di un file fisico di origine che può assumere qualsiasi nome.

Gli esempi si applicano alla DLQ denominata ABC1.DEAD.LETTER.QUEUE, di proprietà del gestore code predefinito.

Se non si specifica la DLQ o il gestore code come mostrato, viene utilizzato il gestore code predefinito per l'installazione insieme alla DLQ appartenente a tale gestore code.

Il comando STRMQMDLQ prende l'input dalla tabella delle regole.

Per eseguire il gestore DLQ, è necessario essere autorizzati ad accedere sia alla DLQ stessa, sia a tutte le code messaggi a cui vengono inoltrati i messaggi sulla DLQ. È inoltre necessaria l'autorizzazione ad assumere l'identità di altri utenti, affinché la DLQ possa inserire messaggi nelle code con l'autorizzazione dell'ID utente nel contesto del messaggio.

### Concetti correlati

[Code di messaggi non recapitabili](#)

### Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

 *La tabella delle regole del gestore DLQ su IBM i*

La tabella delle regole del gestore code di messaggi non recapitabili definisce il modo in cui il gestore DLQ elabora i messaggi che arrivano sulla DLQ IBM i.

La tabella delle regole del gestore DLQ definisce il modo in cui il gestore DLQ elabora i messaggi che arrivano sul DLQ. Esistono due tipi di voci in una tabella di regole:

- La prima voce nella tabella, che è facoltativa, contiene *dati di controllo*.
- Tutte le altre voci nella tabella sono *regole* per il gestore DLQ da seguire. Ciascuna regola è composta da un *pattern* (una serie di caratteristiche del messaggio) rispetto al quale viene confrontato un messaggio e da un' *azione* da intraprendere quando un messaggio sulla DLQ corrisponde al pattern specificato. In una tabella di regole deve essere presente almeno una regola.

Ciascuna voce nella tabella delle regole è composta da una o più parole chiave.

### Dati di controllo

Questa sezione descrive le parole chiave che possono essere incluse in una voce di dati di controllo in una tabella di regole gestore DLQ. Tieni presente quanto segue:

- Il valore predefinito per una parola chiave, se presente, è sottolineato.
- La linea verticale (|) separa le alternative. È possibile specificare solo uno di questi.
- Tutte le parole chiave sono facoltative.

#### **INPUTQ ( *QueueName* | ' ' (predefinito))**

Il nome della DLQ che si desidera elaborare:

1. Qualsiasi valore UDLMMSGQ (o \*DFT) specificato come parametro per il comando **STRMQMDLQ** sostituisce qualsiasi valore INPUTQ nella tabella delle regole.
2. Se si specifica un valore UDLMMSGQ vuoto come parametro per il comando **STRMQMDLQ**, viene utilizzato il valore INPUTQ nella tabella delle regole.
3. Se si specifica un valore UDLMMSGQ vuoto come parametro per il comando **STRMQMDLQ** e un valore INPUTQ vuoto nella tabella delle regole, viene utilizzata la coda di messaggi non recapitabili predefinita di sistema.

#### **INPUTQM ( NomeQueueManager| ' ' (predefinito))**

Il nome del gestore code proprietario della DLQ denominata nella parola chiave INPUTQ.

Se non si specifica un gestore code, oppure si specifica INPUTQM ("") nella tabella delle regole, il sistema utilizza il gestore code predefinito per l'installazione.

#### **RETRYINT ( Intervallo|60 (predefinito))**

L'intervallo, in secondi, con cui il gestore DLQ deve tentare di rielaborare i messaggi sulla DLQ che non è stato possibile elaborare al primo tentativo e per cui sono stati richiesti ripetuti tentativi. Per impostazione predefinita, l'intervallo tra i tentativi è 60 secondi.

#### **WAIT ( YES (predefinito) |NO|nnn )**

Indica se il gestore DLQ deve attendere l'arrivo di ulteriori messaggi sulla DLQ quando rileva che non sono presenti ulteriori messaggi che può elaborare.

##### **SÌ**

Fa sì che il gestore DLQ attenda indefinitamente.

##### **NO**

Causa la chiusura del gestore DLQ quando rileva che il DLQ è vuoto o non contiene messaggi che può elaborare.

##### **nnn**

Fa in modo che il gestore DLQ attenda per *nnn* secondi l'arrivo di nuovo lavoro prima di terminare, dopo che ha rilevato che la coda è vuota o non contiene alcun messaggio che può elaborare.

Specificare WAIT (YES) per le DLQ occupate e WAIT (NO) o WAIT ( *nnn* ) per i DLQ che hanno un livello di attività basso. Se il gestore DLQ può terminare, richiamarlo nuovamente utilizzando il trigger.

È possibile fornire il nome della DLQ come parametro di input al comando **STRMQMDLQ**, come alternativa all'inclusione dei dati di controllo nella tabella delle regole. Se un qualsiasi valore viene specificato sia nella tabella delle regole che nell'input del comando **STRMQMDLQ**, il valore specificato nel comando **STRMQMDLQ** ha la precedenza.

**Nota:** Se una voce di dati di controllo è inclusa nella tabella delle regole, deve essere la prima voce nella tabella.

#### Regole DLQ (modelli e azioni) su IBM i

Una descrizione dei modelli e delle azioni per ciascuna regola della coda di messaggi non instradabili IBM i.

Di seguito è riportata una regola di esempio da una tabella di regole del gestore DLQ:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +
ACTION (RETRY) RETRY (3)
```

Questa regola indica al gestore DLQ di effettuare 3 tentativi di recapitare alla coda di destinazione qualsiasi messaggio persistente inserito nella DLQ perché MQPUT e MQPUT1 non erano consentiti.

Questa sezione descrive le parole chiave che è possibile includere in una regola. Tieni presente quanto segue:

- Il valore predefinito per una parola chiave, se presente, è sottolineato. Per la maggior parte delle parole chiave, il valore predefinito è \* (asterisco), che corrisponde a qualsiasi valore.
- La linea verticale (|) separa le alternative. È possibile specificare solo uno di questi.

- Tutte le parole chiave tranne ACTION sono facoltative.

Questa sezione inizia con una descrizione delle parole chiave corrispondenti al modello (quelle con cui i messaggi sulla DLQ vengono messi in corrispondenza). Descrive quindi le parole chiave dell'azione (quelle che determinano in che modo il gestore DLQ deve elaborare un messaggio corrispondente).

### **IBM i** Modello DLQ - parole chiave corrispondenti su IBM i

Le parole chiave corrispondenti al modello sono descritte in un esempio. Utilizzare queste parole chiave per specificare i valori a cui corrispondono i messaggi sulla coda di messaggi non instradabili IBM i . Tutte le parole chiave corrispondenti al modello sono facoltative.

#### **APPLIDAT ( *ApplIdentityData* | \* (predefinito))**

Il valore *ApplIdentityData* del messaggio sulla DLQ, specificato nel descrittore del messaggio, MQMD.

#### **APPLNAME ( *PutApplName* | \* (predefinito))**

Il nome dell'applicazione che ha emesso la chiamata MQPUT o MQPUT1 , come specificato nel campo *PutApplName* del descrittore del messaggio, MQMD, del messaggio sulla DLQ.

#### **APPLTYPE ( *PutApplTipo* | \* (predefinito))**

Il valore *PutApplType* specificato nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

#### **DESTQ ( *QueueName* | \* (valore predefinito))**

Il nome della coda messaggi a cui è destinato il messaggio.

#### **DESTQM (Nome *QueueManager* | \* (predefinito))**

Il nome del gestore code per la coda messaggi a cui è destinato il messaggio.

#### **FEEDBACK ( *Feedback* | \* (Predefinito))**

Quando il valore *MsgType* è MQMT\_REPORT, *Feedback* descrive la natura del report.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQFB\_COA per identificare i messaggi sulla DLQ che richiedono la conferma del relativo arrivo sulle code di destinazione.

#### **FORMAT ( *Formato* | \* (predefinito))**

Il nome utilizzato dal mittente del messaggio per descrivere il formato dei dati del messaggio.

#### **MSGTYPE ( *MsgType* | \* (valore predefinito))**

Il tipo di messaggio del DLQ.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQMT\_REQUEST per identificare i messaggi sulla DLQ che richiedono risposte.

#### **PERSIST ( *Persistenza* | \* (predefinito))**

Il valore di persistenza del messaggio. La persistenza di un messaggio determina se sopravvive ai riavvii del gestore code.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQPER\_PERSISTENT per identificare quei messaggi nella DLQ che sono persistenti.

#### **REASON ( *ReasonCode* | \* (predefinito))**

Il codice di errore che descrive il motivo per cui il messaggio è stato inserito nella DLQ.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQRC\_Q\_FULL per identificare i messaggi collocati nella DLQ perché le code di destinazione erano piene.

#### **REPLYQ ( *QueueName* | \* (predefinito))**

Il nome della coda di risposta specificato nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

#### **REPLYQM ( *QueueManagerNome* | \* (predefinito))**

Il nome del gestore code della coda di risposta specificato nella parola chiave REPLYQ.

**USERID ( *UserIdentifier* | \* (predefinito))**

L'ID utente dell'utente che ha creato il messaggio sulla DLQ, come specificato nel descrittore del messaggio, MQMD.

 **Parole chiave di azione DLQ su IBM i**

Utilizzare queste parole chiave dell'azione della coda di messaggi non recapitabili per determinare come viene elaborato un messaggio corrispondente sulla coda di messaggi non recapitabili IBM i .

**AZIONE (DISCARD | IGNORE | RETRY | FWD)**

L'azione eseguita per qualsiasi messaggio sul DLQ che corrisponde al pattern definito in questa regola.

**DISCARD**

Causa la cancellazione del messaggio dalla DLQ.

**IGNORE**

Fa sì che il messaggio venga conservato nella DLQ.

**RIPROVA**

Fa in modo che il gestore DLQ tenti nuovamente di inserire il messaggio nella coda di destinazione.

**FWD**

Il gestore DLQ inoltra il messaggio alla coda denominata sulla parola chiave FWDQ.

È necessario specificare la parola chiave ACTION. Il numero di tentativi effettuati per implementare un'azione è gestito dalla parola chiave RETRY. La parola chiave RETRYINT dei dati di controllo controlla l'intervallo tra i tentativi.

**FWDQ ( *QueueName* | & DESTQ | & REPLYQ)**

Il nome della coda messaggi a cui viene inoltrato il messaggio quando si seleziona la parola chiave ACTION.

***QueueName***

Il nome di una coda messaggi. FWDQ ("") non è valido.

**& DESTQ**

Prendere il nome della coda dal campo *DestQName* nella struttura MQDLH.

**& REPLYQ**

Prendere il nome della coda dal campo *ReplyToQ* nel descrittore del messaggio, MQMD.

È possibile specificare REPLYQ (? \*) nel modello di messaggio per evitare messaggi di errore, quando una regola che specifica FWDQ (& REPLYQ) corrisponde a un messaggio con un campo *ReplyToQ* vuoto.

**FWDQM ( *QueueManagerName* | & DESTQM | & REPLYQM | ' ' (predefinito))**

Il gestore code della coda a cui viene inoltrato un messaggio.

***QueueManagerName***

Il nome del gestore code per la coda a cui viene inoltrato il messaggio quando si seleziona la parole chiave ACTION (FWD).

**& DESTQM**

Prendere il nome gestore code dal campo *DestQMGrNome* nella struttura MQDLH.

**& REPLYQM**

Prendere il nome del gestore code dal campo *ReplyToReplyTo* nel descrittore del messaggio, MQMD.

..

FWDQM (""), che è il valore predefinito, identifica il gestore code locale.

**HEADER ( YES (predefinito) | NO)**

Se MQDLH deve rimanere su un messaggio per cui è richiesto ACTION (FWD). Per impostazione predefinita, MQDLH rimane sul messaggio. La parola chiave HEADER non è valida per azioni diverse da FWD.

## PUTAUT ( DEF (predefinito) | CTX)

L'autorità con cui i messaggi devono essere inseriti dal gestore DLQ:

### DEF

Inserisce i messaggi con l'autorità del gestore DLQ.

### CTX

Fa in modo che i messaggi vengano inseriti con l'autorizzazione dell'ID utente nel contesto del messaggio. È necessario essere autorizzati ad assumere l'identità di altri utenti, se si specifica PUTAUT (CTX).

## RETRY ( RetryCount|1 (predefinito)

Il numero di volte, nell'intervallo compreso tra 1 e 999.999.999, per tentare un'azione (all'intervallo specificato nella parola chiave RETRYINT dei dati di controllo).

**Nota:** Il conteggio dei tentativi effettuati dal gestore DLQ per implementare una particolare regola è specifico per l'istanza corrente del gestore DLQ; il conteggio non persiste durante i riavvii. Se si riavvia il gestore DLQ, il numero di tentativi effettuati per applicare una regola viene reimpostato su zero.

## IBM i *Convenzioni della tabella regole DLQ su IBM i*

La tabella delle regole della coda di messaggi non recapitabili IBM i deve rispettare convenzioni specifiche relative alla sintassi, alla struttura e al contenuto.

- Una tabella di regole deve contenere almeno una regola.
- Le parole chiave possono essere presenti in qualsiasi ordine.
- Una parola chiave può essere inclusa una sola volta in qualsiasi regola.
- Le parole chiave non sono sensibili al maiuscolo / minuscolo.
- Una parola chiave e il suo valore di parametro devono essere separati da altre parole chiave da almeno uno spazio o una virgola.
- Qualsiasi numero di spazi vuoti può verificarsi all'inizio o alla fine di una regola e tra parole chiave, punteggiatura e valori.
- Ogni regola deve iniziare su una nuova riga.
- Per la portabilità, la lunghezza significativa di una riga non deve superare i 72 caratteri.
- Utilizzare il segno più (+) come ultimo carattere non vuoto su una riga per indicare che la regola continua dal primo carattere non vuoto nella riga successiva. Utilizzare il segno meno (-) come ultimo carattere non vuoto su una riga per indicare che la regola continua dall'inizio della riga successiva. I caratteri di continuazione possono verificarsi all'interno di parole chiave e parametri.

Ad esempio:

```
APPLNAME('ABC+  
D')
```

risultati in 'ABCD'.

```
APPLNAME('ABC-  
D')
```

risultati in ' ABC D'.

- Le righe di commento, che iniziano con un asterisco (\*), possono trovarsi in qualsiasi punto della tabella delle regole.
- Le righe vuote vengono ignorate.
- Ogni voce nella tabella delle regole del programma di gestione DLQ comprende una o più parole chiave e i relativi parametri associati. I parametri devono seguire queste regole di sintassi:

- Ogni valore di parametro deve includere almeno un carattere significativo. Le virgolette delimitate nei valori racchiusi tra virgolette non sono considerate significative. Ad esempio, questi parametri sono validi:

FORMAT('ABC')	3 caratteri significativi
FORMAT(ABC)	3 caratteri significativi
FORMAT('A')	1 carattere significativo
FORMAT(A)	1 carattere significativo
FORMAT(' ')	1 carattere significativo

Questi parametri non sono validi perché non contengono caratteri significativi:

```
FORMAT(' ')
FORMAT( )
FORMAT( )
FORMAT
```

- sono supportati. È possibile utilizzare il punto interrogativo (?) al posto di qualsiasi carattere singolo, ad eccezione di uno spazio finale. È possibile utilizzare l'asterisco (\*) al posto di zero o più caratteri adiacenti. L'asterisco (\*) e il punto interrogativo (?) sono **sempre** interpretati come caratteri jolly nei parametri.
- Non è possibile includere caratteri jolly nei parametri di queste parole chiave: ACTION, HEADER, RETRY, FWDQ, FWDQM e PUTAUT.
- Gli spazi vuoti finali nei valori di parametro e nei campi corrispondenti nel messaggio sulla DLQ, non sono significativi quando si eseguono corrispondenze di caratteri jolly. Tuttavia, gli spazi vuoti iniziali e incorporati all'interno delle stringhe tra virgolette sono significativi per le corrispondenze dei caratteri jolly.
- I parametri numerici non possono includere il carattere jolly punto interrogativo (?). È possibile includere l'asterisco (\*) al posto di un intero parametro numerico, ma l'asterisco non può essere incluso come parte di un parametro numerico. Ad esempio, questi sono parametri numerici validi:

MSGTYPE(2)	Sono idonei solo i messaggi di risposta
MSGTYPE(*)	Qualsiasi tipo di messaggio è idoneo
MSGTYPE(' *')	Qualsiasi tipo di messaggio è idoneo

Tuttavia, MSGTYPE('2\*') non è valido, poiché include un asterisco (\*) come parte di un parametro numerico.

- I parametri numerici devono essere compresi nell'intervallo 0 - 999 999 999. Se il valore del parametro è compreso in questo intervallo, viene accettato, anche se non è attualmente valido nel campo a cui si riferisce la parola chiave. È possibile utilizzare nomi simbolici per parametri numerici.
- Se un valore stringa è più breve del campo in MQDLH o MQMD a cui si riferisce la parola chiave, il valore viene riempito con spazi vuoti fino alla lunghezza del campo. Se il valore, esclusi gli asterischi, è più lungo del campo, viene diagnosticato un errore. Ad esempio, questi sono tutti valori di stringa validi per un campo di 8 caratteri:

'ABCDEFGH'	8 caratteri
'A*C*E*G*I'	5 caratteri esclusi gli asterischi
'*A*C*E*G*I*K*M*O*'	8 caratteri esclusi gli asterischi

- Le stringhe che contengono spazi, caratteri minuscoli o caratteri speciali diversi da punto (.), barra (?), sottolineatura (\_) e segno di percentuale (%) devono essere racchiusi tra virgolette singole. I caratteri minuscoli non racchiusi tra virgolette vengono piegati in maiuscolo. Se la stringa include

una virgoletta, è necessario utilizzare due virgolette singole per indicare sia l'inizio che la fine della virgoletta. Quando viene calcolata la lunghezza della stringa, ogni ricorrenza delle virgolette viene conteggiata come un singolo carattere.

#### **IBM i** *Elaborazione della tabella regole DLQ su IBM i*

Il gestore code di messaggi non recapitabili ricerca nella tabella delle regole una regola con un modello che corrisponde a un messaggio nella coda di messaggi non recapitabili IBM i .

La ricerca inizia con la prima regola nella tabella e continua in sequenza attraverso la tabella. Quando viene trovata una regola con un modello corrispondente, la tabella delle regole tenta l'azione da tale regola. Il gestore DLQ incrementa il numero di tentativi per una regola di 1 ogni volta che tenta di applicare tale regola. Se il primo tentativo ha esito negativo, il tentativo viene ripetuto fino a quando il conteggio dei tentativi effettuati non corrisponde al numero specificato nella parola chiave RETRY. Se tutti i tentativi hanno esito negativo, il gestore DLQ ricerca la successiva regola di corrispondenza nella tabella.

Questo processo viene ripetuto per le regole di corrispondenza successive fino a quando un'azione ha esito positivo. Quando ogni regola di corrispondenza è stata tentata il numero di volte specificato nella relativa parola chiave RETRY e tutti i tentativi hanno avuto esito negativo, viene assunto ACTION (IGNORE). ACTION (IGNORE) viene utilizzato anche se non viene trovata alcuna regola corrispondente.

#### **Nota:**

1. I modelli di regole corrispondenti vengono ricercati solo per i messaggi sulla DLQ che iniziano con un MQDLH. I messaggi che non iniziano con MQDLH vengono riportati periodicamente come in errore e rimangono nella DLQ per un tempo indefinito.
2. Tutte le parole chiave del modello possono essere predefinite, in modo che una regola possa essere composta solo di un'azione. Tenere presente, tuttavia, che le regole di sola azione vengono applicate a tutti i messaggi nella coda che hanno MQDLH e che non sono già stati elaborati in conformità con altre regole nella tabella.
3. La tabella delle regole viene convalidata all'avvio del programma di gestione DLQ e gli errori vengono contrassegnati in quel momento. (I messaggi di errore emessi dal gestore DLQ sono descritti in [Messaggi e codici di errore](#).) È possibile apportare delle modifiche alla tabella delle regole in qualsiasi momento, ma tali modifiche non diventano effettive fino a quando il gestore DLQ non viene riavviato.
4. Il gestore DLQ non modifica il contenuto dei messaggi, di MQDLH o del descrittore del messaggio. Il gestore DLQ inserisce sempre i messaggi in altre code con l'opzione del messaggio MQPMO\_PASS\_ALL\_CONTEXT.
5. Gli errori di sintassi consecutivi nella tabella delle regole potrebbero non essere riconosciuti, poiché la convalida della tabella delle regole elimina la creazione di errori ripetitivi.
6. Il gestore DLQ apre il DLQ con l'opzione MQOO\_INPUT\_AS\_Q\_DEF.
7. Più istanze del gestore DLQ possono essere eseguite contemporaneamente sulla stessa coda, utilizzando la stessa tabella regole. Tuttavia, è più usuale che ci sia una relazione uno - a - uno tra un gestore DLQ e un gestore DLQ.

#### **IBM i** *Verifica che tutti i messaggi DLQ siano elaborati su IBM i*

Il gestore code di messaggi non instradabili conserva un record di tutti i messaggi sulla DLQ IBM i che sono stati visualizzati ma non rimossi.

Se si utilizza il gestore DLQ come filtro per estrarre un piccolo sottoinsieme di messaggi dal DLQ, il gestore DLQ conserva ancora un record di tali messaggi sul DLQ che non ha elaborato. Inoltre, il gestore DLQ non può garantire che i nuovi messaggi in arrivo sul DLQ vengano visualizzati, anche se il DLQ è definito come FIFO (first - in first - out). Se la coda non è vuota, la DLQ viene periodicamente riesaminata per controllare tutti i messaggi.

Per questi motivi, assicurarsi che la DLQ contenga il minor numero di messaggi possibile. Se i messaggi che non possono essere eliminati o inoltrati ad altre code (per qualsiasi motivo) possono accumularsi sulla coda, il carico di lavoro del gestore DLQ aumenta e il DLQ stesso rischia di riempirsi.

È possibile adottare misure specifiche per abilitare il gestore DLQ a svuotarlo. Ad esempio, provare a non utilizzare ACTION (IGNORE), che lascia i messaggi sulla DLQ. (Ricordare che ACTION (IGNORE) viene assunto per i messaggi che non sono esplicitamente indirizzati da altre regole nella tabella.) Invece, per quei messaggi che altrimenti verrebbero ignorati, utilizzare un'azione che sposta i messaggi su un'altra coda. Ad esempio:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Allo stesso modo, rendere la regola finale nella tabella un catchall per elaborare i messaggi che non sono stati indirizzati dalle regole precedenti nella tabella. Ad esempio, la regola finale nella tabella potrebbe essere simile alla seguente:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Ciò fa sì che i messaggi che rientrano nella regola finale nella tabella vengano inoltrati alla coda REALLY.DEAD.QUEUE, dove possono essere elaborati manualmente. Se non si dispone di una regola di questo tipo, è probabile che i messaggi rimangano nella DLQ per un periodo di tempo indefinito.

**IBM i**

*Una tabella di regole del gestore DLQ di esempio su IBM i*

Codice di esempio per una tabella di regole del gestore code di messaggi non instradabili su IBM i. Questa tabella di regole di esempio contiene una singola immissione dati di controllo e diverse regole.

```
*****
*   An example rules table for the STRMQMDLQ command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* STRMQMDLQ, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to STRMQMDLQ,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation is always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never does things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

```
* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.
```

```
REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it must be able
* to cope with the message being lost, so we can afford to
* discard the message.
```

```
PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
```

```
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We do not have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

## ***Richiamo del gestore DLQ***

Richiamare il gestore code di messaggi non recapitabili utilizzando il comando `runmqdlq`. È possibile denominare la DLQ che si desidera elaborare e il gestore code che si desidera utilizzare in due modi.

I due modi sono i seguenti:

- Come parametri per `runmqdlq` dal prompt dei comandi. Ad esempio:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrul.e.rul
```

- Nella tabella delle regole. Ad esempio:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Gli esempi si applicano al DLQ denominato `ABC1.DEAD.LETTER.QUEUE`, di proprietà del gestore code `ABC1.QUEUE.MANAGER`.

Se non si specifica la DLQ o il gestore code come mostrato, viene utilizzato il gestore code predefinito per l'installazione insieme alla DLQ appartenente a tale gestore code.

Il comando `runmqdlq` prende il suo input da `stdin`; si associa la tabella delle regole a `runmqdlq` reindirizzando `stdin` dalla tabella delle regole.

Per eseguire il gestore DLQ è necessario essere autorizzati ad accedere sia alla DLQ stessa che a tutte le code messaggi a cui vengono inoltrati i messaggi sulla DLQ. Affinché il gestore DLQ inserisci i messaggi nelle code con l'autorizzazione dell'ID utente nel contesto del messaggio, è necessario essere autorizzati anche ad assumere l'identità di altri utenti.

Per ulteriori informazioni sul comando `runmqdlq`, consultare [runmqdlq](#).

## **Concetti correlati**

[Code di messaggi non recapitabili](#)

## Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

### Il gestore DLQ di esempio **amqsdlq**

Oltre al gestore code di messaggi non recapitabili richiamato utilizzando il comando **runmqdlq**, IBM MQ fornisce l'origine di un gestore DLQ di esempio **amqsdlq** con una funzione simile a quella fornita da **runmqdlq**.

È possibile personalizzare **amqsdlq** per fornire un gestore DLQ che soddisfi le proprie esigenze. Ad esempio, è possibile decidere che si desidera un gestore DLQ che possa elaborare i messaggi senza intestazioni di messaggi non recapitabili. (Sia il gestore DLQ predefinito che l'esempio, **amqsdlq**, elaborano solo i messaggi sul DLQ che iniziano con un'intestazione di lettera non recapitabile, MQDLH. I messaggi che non iniziano con un MQDLH vengono identificati come errati e rimangono sulla DLQ per un tempo indefinito.)

`MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ.

In IBM MQ for Windows, l'origine di **amqsdlq** viene fornita nella directory:

```
MQ_INSTALLATION_PATH\tools\c\samples\dlq
```

e la versione compilata viene fornita nella directory:

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

In sistemi IBM MQ for UNIX e Linux, l'origine di **amqsdlq** viene fornita nella directory:

```
MQ_INSTALLATION_PATH/samp/dlq
```

e la versione compilata viene fornita nella directory:

```
MQ_INSTALLATION_PATH/samp/bin
```

**V 9.2.3** Viene inclusa una versione integrata del programma di esempio, denominata **amqsdlqc**. È possibile utilizzare questa opzione per connettersi a un gestore code remoto in modalità client. Per utilizzare **amqsdlqc**, è necessario impostare una delle variabili di ambiente **MQSERVER**, **MQCHLLIB** o **MQCHLTAB** per identificare la modalità di connessione al gestore code. Ad esempio:

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414) "
```

## Tabella regole gestore DLQ

La tabella delle regole del gestore code di messaggi non instradabili definisce il modo in cui il gestore DLQ elabora i messaggi che arrivano sulla DLQ.

Esistono due tipi di voci in una tabella di regole:

- La prima voce nella tabella, che è facoltativa, contiene *dati di controllo*.
- Tutte le altre voci nella tabella sono *regole* per il gestore DLQ da seguire. Ciascuna regola è composta da un *pattern* (una serie di caratteristiche del messaggio) rispetto al quale viene confrontato un messaggio e da un' *azione* da intraprendere quando un messaggio sulla DLQ corrisponde al pattern specificato. In una tabella di regole deve essere presente almeno una regola.

Ciascuna voce nella tabella delle regole è composta da una o più parole chiave.

## Concetti correlati

[Code di messaggi non recapitabili](#)

## Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

### *Dati di controllo DLQ*

È possibile includere parole chiave in una voce di dati di controllo in una tabella di regole del gestore code di messaggi non recapitabili.

#### **Nota:**

- La linea verticale (|) separa le alternative, solo una delle quali può essere specificata.
- Tutte le parole chiave sono facoltative.

#### **INPUTQ ( *QueueName* | ' ' (predefinito) )**

Il nome della DLQ che si desidera elaborare:

1. Qualsiasi valore INPUTQ fornito come parametro per il comando `runmqdlq` sovrascrive qualsiasi valore INPUTQ nella tabella delle regole.
2. Se non si specifica un valore INPUTQ come parametro per il comando `runmqdlq`, ma si **specifica** un valore nella tabella delle regole, viene utilizzato il valore INPUTQ nella tabella delle regole.
3. Se non viene specificato alcun DLQ o si specifica INPUTQ ('') nella tabella delle regole, viene utilizzato il nome del DLQ appartenente al gestore code con il nome fornito come parametro per il comando `runmqdlq`.
4. Se non si specifica un valore INPUTQ come parametro per il comando `runmqdlq` o come valore nella tabella delle regole, viene utilizzato il DLQ appartenente al gestore code denominato sulla parola chiave INPUTQM nella tabella delle regole.

#### **INPUTQM ( *NomeQueueManager* | ' ' (predefinito) )**

Il nome del gestore code che possiede il DLQ denominato nella parola chiave INPUTQ:

1. Qualsiasi valore INPUTQM fornito come parametro al comando `runmqdlq` sovrascrive qualsiasi valore INPUTQM nella tabella delle regole.
2. Se non si specifica un valore INPUTQM come parametro del comando `runmqdlq`, viene utilizzato il valore INPUTQM nella tabella delle regole.
3. Se non viene specificato alcun gestore code oppure se si specifica INPUTQM ('') nella tabella delle regole, viene utilizzato il gestore code predefinito per l'installazione.

#### **RETRYINT ( *Intervallo* | 60 (predefinito) )**

L'intervallo, in secondi, con cui il gestore DLQ deve rielaborare i messaggi sulla DLQ che non è stato possibile elaborare al primo tentativo e per cui sono stati richiesti tentativi ripetuti. Per impostazione predefinita, l'intervallo tra i tentativi è 60 secondi.

#### **WAIT ( YES (predefinito) | NO | *nnn* )**

Indica se il gestore DLQ deve attendere l'arrivo di ulteriori messaggi sulla DLQ quando rileva che non sono presenti ulteriori messaggi che può elaborare.

##### **Sì**

Il gestore DLQ attende indefinitamente.

##### **NO**

Il gestore DLQ termina quando rileva che il DLQ è vuoto o non contiene messaggi che può elaborare.

##### ***nnn***

Il gestore DLQ attende per *nnn* secondi l'arrivo di nuovo lavoro prima di terminare, dopo aver rilevato che la coda è vuota o non contiene messaggi che può elaborare.

Specificare WAIT (YES) per le DLQ occupate e WAIT (NO) o WAIT ( *nnn* ) per i DLQ che hanno un livello di attività basso. Se il gestore DLQ è autorizzato a terminare, richiamarlo di nuovo utilizzando il trigger. Per ulteriori informazioni sull'attivazione, consultare [Avvio delle applicazioni IBM MQ utilizzando i trigger](#).

Un'alternativa all'inclusione dei dati di controllo nella tabella delle regole consiste nel fornire i nomi della DLQ e del relativo gestore code come parametri di input per il comando `runmqdlq`. Se si specifica un valore sia nella tabella delle regole che come input per il comando `runmqdlq`, il valore specificato nel comando `runmqdlq` ha la precedenza.

Se si include una voce di dati di controllo nella tabella delle regole, deve essere la **prima** voce nella tabella.

### *Regole DLQ (modelli e azioni)*

Una descrizione delle parole chiave corrispondenti al modello (quelle rispetto alle quali i messaggi sulla coda di messaggi non recapitabili corrispondono) e le parole chiave dell'azione (quelle che determinano in che modo il gestore DLQ deve elaborare un messaggio corrispondente). Viene fornita anche una regola di esempio.

## **Le parole chiave corrispondenti al modello**

Le parole chiave di corrispondenza del modello, che si utilizzano per specificare i valori rispetto ai quali i messaggi sulla DLQ corrispondono, sono le seguenti. (Tutte le parole chiave corrispondenti al modello sono facoltative):

### **APPLIDAT ( *ApplIdentityData* | \* (predefinito))**

Il valore *ApplIdentityData* specificato nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

### **APPLNAME ( *PutApplName* | \* (predefinito))**

Il nome dell'applicazione che ha emesso la chiamata MQPUT o MQPUT1 , come specificato nel campo *PutApplName* del descrittore del messaggio, MQMD, del messaggio sulla DLQ.

### **APPLTYPE ( *PutApplTipo* | \* (predefinito))**

Il valore *PutApplType* , specificato nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

### **DESTQ ( *QueueName* | \* (valore predefinito))**

Il nome della coda messaggi a cui è destinato il messaggio.

### **DESTQM (Nome *QueueManager* | \* (predefinito))**

Il nome del gestore code della coda messaggi a cui è destinato il messaggio.

### **FEEDBACK ( *Feedback* | \* (Predefinito))**

Quando il valore *MsgType* è MQFB\_REPORT, *Feedback* descrive la natura del report.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQFB\_COA per identificare i messaggi sulla DLQ che necessitano di conferma del relativo arrivo sulle code di destinazione.

### **FORMAT ( *Formato* | \* (predefinito))**

Il nome utilizzato dal mittente del messaggio per descrivere il formato dei dati del messaggio.

### **MSGTYPE ( *MsgType* | \* (valore predefinito))**

Il tipo di messaggio del DLQ.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQMT\_REQUEST per identificare i messaggi sulla DLQ che richiedono risposte.

### **PERSIST ( *Persistenza* | \* (predefinito))**

Il valore di persistenza del messaggio. La persistenza di un messaggio determina se sopravvive ai riavvii del gestore code.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQPER\_PERSISTENT per identificare i messaggi sulla DLQ persistenti.

### **REASON ( *ReasonCode* | \* (predefinito))**

Il codice di errore che descrive il motivo per cui il messaggio è stato inserito nella DLQ.

È possibile utilizzare nomi simbolici. Ad esempio, è possibile utilizzare il nome simbolico MQRC\_Q\_FULL per identificare i messaggi collocati nella DLQ perché le code di destinazione erano piene.

### **REPLYQ ( *QueueName* | \* (predefinito))**

Il nome della coda di risposta specificata nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

**REPLYQM ( *QueueManagerNome* | \* (predefinito) )**

Il nome del gestore code della coda di risposta, come specificato nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

**USERID ( *UserIdentifier* | \* (predefinito) )**

L'ID utente dell'utente che ha creato il messaggio sulla DLQ, come specificato nel descrittore del messaggio, MQMD, del messaggio sulla DLQ.

**Le parole chiave di azione**

Le parole chiave dell'azione, utilizzate per descrivere come deve essere elaborato un messaggio corrispondente, sono le seguenti:

**AZIONE ( DISCARD | IGNORE | RETRY | FWD )**

L'azione da intraprendere per qualsiasi messaggio sulla DLQ che corrisponde al modello definito in questa regola.

**DISCARD**

Cancellare il messaggio dalla DLQ.

**IGNORE**

Lasciare il messaggio sulla DLQ.

**RIPROVA**

Se il primo tentativo di inserire il messaggio nella relativa coda di destinazione ha esito negativo, riprovare. La parola chiave RETRY imposta il numero di tentativi effettuati per implementare un'azione. La parola chiave RETRYINT dei dati di controllo controlla l'intervallo tra i tentativi.

**FWD**

Inoltrare il messaggio alla coda indicata nella parola chiave FWDQ.

È necessario specificare la parola chiave ACTION.

**FWDQ ( *QueueName* | & DESTQ | & REPLYQ )**

Il nome della coda messaggi a cui inoltrare il messaggio quando viene richiesto ACTION (FWD).

***QueueName***

Il nome di una coda messaggi. FWDQ ("") non è valido.

**& DESTQ**

Prendere il nome della coda dal *campoDestQName* nella struttura MQDLH.

**& REPLYQ**

Prendere il nome della coda dal campo *ReplyToQ* nel descrittore del messaggio, MQMD.

Per evitare messaggi di errore quando una regola che specifica FWDQ (& REPLYQ) corrisponde ad un messaggio con un campo *ReplyToQ* vuoto, specificare REPLYQ (? \*) nel modello del messaggio.

**FWDQM ( *QueueManagerName* | & DESTQM | & REPLYQM | ' ' (predefinito) )**

Il gestore code della coda a cui inoltrare un messaggio.

***QueueManagerName***

Il nome del gestore code della coda a cui inoltrare un messaggio quando viene richiesto ACTION (FWD).

**& DESTQM**

Prendere il nome gestore code dal campo *DestQMgrNome* nella struttura MQDLH.

**& REPLYQM**

Prendere il nome del gestore code dal campo *ReplyToReplyTo* nel descrittore del messaggio, MQMD.

''

FWDQM (""), che è il valore predefinito, identifica il gestore code locale.

### **HEADER ( YES (predefinito) |NO)**

Se MQDLH deve rimanere su un messaggio per cui è richiesto ACTION (FWD). Per impostazione predefinita, MQDLH rimane sul messaggio. La parola chiave HEADER non è valida per azioni diverse da FWD.

### **PUTAUT ( DEF (predefinito) | CTX)**

L'autorità con cui i messaggi devono essere inseriti dal gestore DLQ:

#### **DEF**

Inserire i messaggi con l'autorità del gestore DLQ.

#### **CTX**

Inserire i messaggi con l'autorizzazione dell'ID utente nel contesto del messaggio. Se si specifica PUTAUT (CTX), è necessario essere autorizzati ad assumere l'identità di altri utenti.

### **RETRY ( RetryCount|1 (predefinito)**

Il numero di volte, nell'intervallo compreso tra 1 e 999.999.999, per tentare un'azione (all'intervallo specificato nella parola chiave RETRYINT dei dati di controllo). Il conteggio dei tentativi effettuati dal gestore DLQ per implementare una particolare regola è specifico per l'istanza corrente del gestore DLQ; il conteggio non persiste durante i riavvii. Se il gestore DLQ viene riavviato, il numero di tentativi effettuati per applicare una regola viene reimpostato su zero.

## **Regola di esempio**

Di seguito è riportata una regola di esempio da una tabella di regole del gestore DLQ:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Questa regola indica al gestore DLQ di effettuare tre tentativi per consegnare alla relativa coda di destinazione qualsiasi messaggio persistente inserito nella DLQ perché MQPUT e MQPUT1 non erano consentiti.

Tutte le parole chiave che puoi utilizzare su una regola sono descritte nel resto di questa sezione. Tieni presente quanto segue:

- Il valore predefinito per una parola chiave, se presente, è sottolineato. Per la maggior parte delle parole chiave, il valore predefinito è \* (asterisco), che corrisponde a qualsiasi valore.
- La linea verticale (|) separa le alternative, solo una delle quali può essere specificata.
- Tutte le parole chiave tranne ACTION sono facoltative.

#### *Convenzioni della tabella delle regole DLQ*

La sintassi, la struttura e il contenuto della tabella delle regole del gestore code di messaggi non recapitabili devono rispettare queste convenzioni.

La tabella delle regole deve rispettare le convenzioni seguenti:

- Una tabella di regole deve contenere almeno una regola.
- Le parole chiave possono essere presenti in qualsiasi ordine.
- Una parola chiave può essere inclusa solo una volta in una regola.
- Le parole chiave non sono sensibili al maiuscolo / minuscolo.
- Una parola chiave e il suo valore di parametro devono essere separati da altre parole chiave da almeno uno spazio o una virgola.
- Ci può essere qualsiasi numero di spazi vuoti all'inizio o alla fine di una regola e tra parole chiave, punteggiatura e valori.
- Ogni regola deve iniziare su una nuova riga.
- Su sistemi Windows , l'ultima regola della tabella deve terminare con un carattere di ritorno a capo / avanzamento riga. È possibile ottenere ciò assicurandosi di premere il tasto Invio alla fine della regola, in modo che l'ultima riga della tabella sia una riga vuota.

- Per motivi di portabilità, la lunghezza significativa di una riga non deve essere superiore a 72 caratteri.
- Utilizzare il segno più (+) come ultimo carattere non vuoto su una linea per indicare che la regola continua dal primo carattere non vuoto nella riga successiva. Utilizzare il segno meno (-) come ultimo carattere non vuoto su una riga per indicare che la regola continua dall'inizio della riga successiva. I caratteri di continuazione possono verificarsi all'interno di parole chiave e parametri.

Ad esempio:

```
APPLNAME('ABC+
D')
```

si traduce in "ABCD", e

```
APPLNAME('ABC-
D')
```

risultati in 'ABC D'.

- Le righe di commento, che iniziano con un asterisco (\*), possono trovarsi in qualsiasi punto della tabella delle regole.
- Le righe vuote vengono ignorate.
- Ogni voce nella tabella delle regole del programma di gestione DLQ comprende una o più parole chiave e i relativi parametri associati. I parametri devono seguire queste regole di sintassi:
  - Ogni valore di parametro deve includere almeno un carattere significativo. Le virgolette singole delimitate in valori racchiusi tra virgolette non sono considerate significative. Ad esempio, questi parametri sono validi:

FORMAT('ABC')	3 caratteri significativi
FORMAT(ABC)	3 caratteri significativi
FORMAT('A')	1 carattere significativo
FORMAT(A)	1 carattere significativo
FORMAT(' ')	1 carattere significativo

Questi parametri non sono validi perché non contengono caratteri significativi:

```
FORMAT(' ')
FORMAT( )
FORMAT()
FORMAT
```

- sono supportati. È possibile utilizzare il punto interrogativo (?) invece di qualsiasi carattere singolo, tranne uno spazio finale; è possibile utilizzare l'asterisco (\*) invece di zero o più caratteri adiacenti. L'asterisco (\*) e il punto interrogativo (?) sono **sempre** interpretati come caratteri jolly nei parametri.
- I caratteri jolly non possono essere inclusi nei parametri di queste parole chiave: ACTION, HEADER, RETRY, FWDQ, FWDQM e PUTAUT.
- Gli spazi vuoti finali nei valori di parametro e nei campi corrispondenti nel messaggio sulla DLQ, non sono significativi quando si eseguono corrispondenze di caratteri jolly. Tuttavia, gli spazi vuoti iniziali e incorporati nelle stringhe racchiusi tra virgolette singole sono significativi per le corrispondenze di caratteri jolly.
- I parametri numerici non possono includere il carattere jolly punto interrogativo (?). È possibile utilizzare l'asterisco (\*) invece di un intero parametro numerico, ma non come parte di un parametro numerico. Ad esempio, questi sono parametri numerici validi:

MSGTYPE(2)	Sono idonei solo i messaggi di risposta
MSGTYPE(*)	Qualsiasi tipo di messaggio è idoneo
MSGTYPE(' * ')	Qualsiasi tipo di messaggio è idoneo

Tuttavia, MSGTYPE(' 2\* ') non è valido, poiché include un asterisco (\*) come parte di un parametro numerico.

- I parametri numerici devono essere compresi nell'intervallo 0 - 999 999 999. Se il valore del parametro è compreso in questo intervallo, viene accettato, anche se non è attualmente valido nel campo a cui si riferisce la parola chiave. È possibile utilizzare nomi simbolici per parametri numerici.
- Se un valore stringa è più breve del campo in MQDLH o MQMD a cui si riferisce la parola chiave, il valore viene riempito con spazi vuoti fino alla lunghezza del campo. Se il valore, esclusi gli asterischi, è più lungo del campo, viene diagnosticato un errore. Ad esempio, questi sono tutti valori di stringa validi per un campo di 8 caratteri:

'ABCDEFGH'	8 caratteri
'A*C*E*G*I'	5 caratteri esclusi gli asterischi
'*A*C*E*G*I*K*M*O *'	8 caratteri esclusi gli asterischi

- Racchiudere le stringhe che contengono spazi, caratteri minuscoli o caratteri speciali diversi da punto (.), barra (?), carattere di sottolineatura ( \_ ) e segno di percentuale (%) tra virgolette singole. I caratteri minuscoli non racchiusi tra virgolette singole vengono piegati in maiuscolo. Se la stringa include una virgoletta, utilizzare due virgolette singole per indicare sia l'inizio che la fine della virgoletta. Quando viene calcolata la lunghezza della stringa, ogni ricorrenza delle virgolette viene conteggiata come un singolo carattere.

### **Modalità di elaborazione della tabella delle regole DLQ**

Il gestore code di messaggi non instradabili ricerca nella tabella delle regole una regola in cui il pattern corrisponde a un messaggio nella DLQ.

La ricerca inizia con la prima regola nella tabella e continua in sequenza attraverso la tabella. Quando il gestore DLQ trova una regola con un modello corrispondente, prende l'azione da tale regola. Il gestore DLQ incrementa il numero di tentativi per una regola di 1 ogni volta che applica tale regola. Se il primo tentativo ha esito negativo, il gestore DLQ tenta di nuovo fino a quando il numero di tentativi non corrisponde al numero specificato nella parola chiave RETRY. Se tutti i tentativi hanno esito negativo, il gestore DLQ ricerca la successiva regola di corrispondenza nella tabella.

Questo processo viene ripetuto per le regole di corrispondenza successive fino a quando un'azione ha esito positivo. Quando ogni regola di corrispondenza è stata tentata il numero di volte specificato nella relativa parola chiave RETRY e tutti i tentativi hanno avuto esito negativo, viene assunto ACTION (IGNORE). ACTION (IGNORE) viene utilizzato anche se non viene trovata alcuna regola corrispondente.

#### **Nota:**

1. I modelli di regole corrispondenti vengono ricercati solo per i messaggi sulla DLQ che iniziano con un MQDLH. I messaggi che non iniziano con MQDLH vengono riportati periodicamente come in errore e rimangono nella DLQ per un tempo indefinito.
2. Tutte le parole chiave del modello possono essere impostate come predefinite, in modo che una regola possa consistere solo in un'azione. Tenere presente, tuttavia, che le regole di sola azione vengono applicate a tutti i messaggi nella coda che hanno MQDLH e che non sono già stati elaborati in conformità con altre regole nella tabella.
3. La tabella di regole viene convalidata all'avvio del gestore DLQ e gli errori vengono contrassegnati in quel momento. È possibile apportare modifiche alla tabella delle regole in qualsiasi momento, ma tali modifiche non diventano effettive fino al riavvio del gestore DLQ.

4. Il gestore DLQ non modifica il contenuto dei messaggi, MQDLH o il descrittore del messaggio. Il gestore DLQ inserisce sempre i messaggi in altre code con l'opzione del messaggio MQPMO\_PASS\_ALL\_CONTEXT.
5. Gli errori di sintassi consecutivi nella tabella delle regole potrebbero non essere riconosciuti perché la tabella delle regole è progettata per eliminare la creazione di errori ripetitivi durante la convalida.
6. Il gestore DLQ apre il DLQ con l'opzione MQOO\_INPUT\_AS\_Q\_DEF.
7. Più istanze del gestore DLQ possono essere eseguite contemporaneamente sulla stessa coda, utilizzando la stessa tabella regole. Tuttavia, è più usuale che ci sia una relazione uno - a - uno tra un gestore DLQ e un gestore DLQ.

### Concetti correlati

[Code di messaggi non recapitabili](#)

### Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

#### Verifica dell'elaborazione di tutti i messaggi DLQ

Il gestore code di messaggi non instradabili conserva un record di tutti i messaggi presenti nella DLQ che sono stati visualizzati ma non rimossi.

Se si utilizza il gestore DLQ come filtro per estrarre un piccolo sottoinsieme di messaggi dal DLQ, il gestore DLQ deve ancora conservare un record di tali messaggi sul DLQ che non ha elaborato. Inoltre, il gestore DLQ non può garantire che vengano visualizzati nuovi messaggi in arrivo sul DLQ, anche se il DLQ è definito come FIFO (first - in - first - out). Se la coda non è vuota, la DLQ viene periodicamente riesaminata per controllare tutti i messaggi.

Per questi motivi, provare a garantire che il DLQ contenga il minor numero di messaggi possibile; se i messaggi che non possono essere scartati o inoltrati ad altre code (per qualsiasi motivo) possono accumularsi sulla coda, il carico di lavoro del gestore DLQ aumenta e il DLQ stesso può riempirsi.

È possibile adottare misure specifiche per abilitare il gestore DLQ a svuotarlo. Ad esempio, provare a non utilizzare ACTION (IGNORE), che lascia i messaggi sulla DLQ. (Ricordare che ACTION (IGNORE) viene assunto per i messaggi che non sono esplicitamente indirizzati da altre regole nella tabella.) Invece, per quei messaggi che altrimenti verrebbero ignorati, utilizzare un'azione che sposta i messaggi su un'altra coda. Ad esempio:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

Allo stesso modo, rendere la regola finale nella tabella un catchall per elaborare i messaggi che non sono stati indirizzati dalle regole precedenti nella tabella. Ad esempio, la regola finale nella tabella potrebbe essere simile alla seguente:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Questo inoltra i messaggi che rientrano nella regola finale nella tabella alla coda REALLY . DEAD . QUEUE, dove possono essere elaborati manualmente. Se non si dispone di una regola di questo tipo, è probabile che i messaggi rimangano nella DLQ per un periodo di tempo indefinito.

### Una tabella di regole gestore DLQ di esempio

Una tabella di regole della coda di messaggi non instradabili di esempio per il comando **runmqdlq**, contenente una singola immissione di dati di controllo e diverse regole.

```
*****
*   An example rules table for the runmqdlq command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
```

```

* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

```

```

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.

REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)

```

## Concetti correlati

[Code di messaggi non recapitabili](#)

## Attività correlate

[Risoluzione dei problemi dei messaggi non recapitati](#)

## Riferimenti correlati

[runmqdlq \(esecuzione gestore code di messaggi non instradabili\)](#)

## Utilizzo degli argomenti di gestione

Utilizzare i comandi MQSC per gestire gli argomenti di amministrazione.

Consultare [Comandi MQSC](#) per informazioni dettagliate su questi comandi.

### Concetti correlati

[Oggetti argomento di gestione](#)

[“Definizione di un argomento di gestione” a pagina 168](#)

Utilizzare il comando MQSC **DEFINE TOPIC** per creare un argomento di gestione. Quando si definisce un argomento di gestione, è possibile facoltativamente impostare ciascun attributo dell'argomento.

[“Visualizzazione degli attributi dell'oggetto argomento di gestione” a pagina 169](#)

Utilizzare il comando MQSC **DISPLAY TOPIC** per visualizzare un oggetto argomento di gestione.

[“Modifica degli attributi dell'argomento di amministrazione” a pagina 169](#)

È possibile modificare gli attributi dell'argomento in due modi, utilizzando il comando **ALTER TOPIC** o il comando **DEFINE TOPIC** con l'attributo **REPLACE**.

[“Copia di una definizione di argomento di amministrazione” a pagina 170](#)

È possibile copiare una definizione di argomento utilizzando l'attributo **LIKE** sul comando **DEFINE**.

[“Eliminazione di una definizione di argomento di gestione” a pagina 170](#)

È possibile utilizzare il comando MQSC **DELETE TOPIC** per eliminare un argomento di amministrazione.

## Definizione di un argomento di gestione

Utilizzare il comando MQSC **DEFINE TOPIC** per creare un argomento di gestione. Quando si definisce un argomento di gestione, è possibile facoltativamente impostare ciascun attributo dell'argomento.

Qualsiasi attributo dell'argomento non esplicitamente impostato viene ereditato dall'argomento di gestione predefinito, SYSTEM.DEFAULT.TOPIC, che è stato creato quando è stata installata l'installazione del sistema.

Ad esempio, il comando **DEFINE TOPIC** che segue, definisce un argomento denominato ORANGE . TOPIC con queste caratteristiche:

- Si risolve nella stringa di argomenti ORANGE. Per informazioni su come utilizzare le stringhe argomento, consultare [Combinazione di stringhe argomento](#).
- Qualsiasi attributo impostato su ASPARENT utilizza l'attributo come definito dall'argomento principale di questo argomento. Questa operazione viene ripetuta nella struttura ad albero degli argomenti fino all'argomento principale, SYSTEM.BASE.TOPIC trovato. Per ulteriori informazioni, consultare [Alberi degli argomenti](#).

```
DEFINE TOPIC (ORANGE.TOPIC) +  
TOPICSTR (ORANGE) +  
DEFPRTY(ASPARENT) +  
NPMSGDLV (ASPARENT)
```

### Nota:

- Tranne che per il valore della stringa di argomenti, tutti i valori di attributo visualizzati sono i valori predefiniti. Sono mostrati qui solo come un'illustrazione. È possibile ometterli se si è certi che i valori predefiniti sono quelli desiderati o non sono stati modificati. Consultare anche [“Visualizzazione degli attributi dell'oggetto argomento di gestione” a pagina 169](#).
- Se si dispone già di un argomento di gestione sullo stesso gestore code con il nome ORANGE.TOPIC, questo comando non riesce. Utilizzare l'attributo **REPLACE** se si desidera sovrascrivere la definizione esistente di un argomento, ma consultare anche [“Modifica degli attributi dell'argomento di amministrazione” a pagina 169](#).

## Riferimenti correlati

[DEFINISCI ARGOMENTO](#)

## Visualizzazione degli attributi dell'oggetto argomento di gestione

Utilizzare il comando MQSC **DISPLAY TOPIC** per visualizzare un oggetto argomento di gestione.

Per visualizzare tutti gli argomenti, utilizzare:

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

È possibile visualizzare selettivamente gli attributi specificandoli singolarmente con il comando **DISPLAY TOPIC**. Ad esempio:

```
DISPLAY TOPIC(ORANGE.TOPIC) +  
TOPICSTR +  
DEFPRTY +  
NPMSGDLV
```

Questo comando visualizza i tre attributi specificati nel modo seguente:

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC)                                TYPE(LOCAL)  
TOPICSTR(ORANGE)                                    DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

Per visualizzare l'argomento ASPARENT come vengono utilizzati al runtime, utilizzare il comando **DISPLAY TPSTATUS**. Ad esempio, utilizzare:

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

Il comando visualizza i seguenti dettagli:

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE)                                DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

Quando si definisce un argomento di gestione, vengono utilizzati tutti gli attributi non specificati esplicitamente dall'argomento di gestione predefinito, denominato SYSTEM.DEFAULT.TOPIC. Per visualizzare questi attributi predefiniti, utilizzare il seguente comando:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

## Riferimenti correlati

[VISUALIZZA ARGOMENTO](#)

[VISUALIZZA TPSTATUS](#)

## Modifica degli attributi dell'argomento di amministrazione

È possibile modificare gli attributi dell'argomento in due modi, utilizzando il comando **ALTER TOPIC** o il comando **DEFINE TOPIC** con l'attributo **REPLACE**.

Se, ad esempio, si desidera modificare la priorità predefinita dei messaggi consegnati ad un argomento denominato ORANGE.TOPIC, deve essere 5, utilizzare uno dei seguenti comandi.

- Utilizzando il comando **ALTER**:

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Questo comando modifica un singolo attributo, quello della priorità predefinita del messaggio consegnato a questo argomento in 5; tutti gli altri attributi rimangono gli stessi.

- Utilizzando il comando **DEFINE** :

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Questo comando modifica la priorità predefinita dei messaggi consegnati a questo argomento. A tutti gli altri attributi vengono assegnati valori predefiniti.

Se si modifica la priorità dei messaggi inviati a questo argomento, i messaggi esistenti non vengono influenzati. Qualsiasi nuovo messaggio, tuttavia, utilizza la priorità specificata se non fornita dall'applicazione di pubblicazione.

### Riferimenti correlati

[ALTER TOPIC](#)

[VISUALIZZA ARGOMENTO](#)

## Copia di una definizione di argomento di amministrazione

È possibile copiare una definizione di argomento utilizzando l'attributo LIKE sul comando **DEFINE** .

Ad esempio:

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Questo comando crea un argomento, MAGENTA.TOPIC, con gli stessi attributi dell'argomento originale, ORANGE.TOPIC, piuttosto che quelli dell'argomento di gestione predefinito del sistema. Immettere il nome dell'argomento da copiare esattamente come è stato immesso quando è stato creato l'argomento. Se il nome contiene caratteri minuscoli, racchiuderlo tra virgolette singole.

È anche possibile utilizzare questo modulo del comando **DEFINE** per copiare una definizione di argomento, ma apportare modifiche agli attributi dell'originale. Ad esempio:

```
DEFINE TOPIC(BLUE.TOPIC) +  
TOPICSTR(BLUE) +  
LIKE(ORANGE.TOPIC)
```

È anche possibile copiare gli attributi dell'argomento BLUE.TOPIC all'argomento GREEN.TOPIC e specificare che quando le pubblicazioni non possono essere consegnate alla coda del sottoscrittore corretta, non vengono inserite nella coda di messaggi non recapitabili. Ad esempio:

```
DEFINE TOPIC(GREEN.TOPIC) +  
TOPICSTR(GREEN) +  
LIKE(BLUE.TOPIC) +  
USEDLQ(NO)
```

### Riferimenti correlati

[DEFINISCI ARGOMENTO](#)

## Eliminazione di una definizione di argomento di gestione

È possibile utilizzare il comando MQSC **DELETE TOPIC** per eliminare un argomento di amministrazione.

Ad esempio:

```
DELETE TOPIC(ORANGE.TOPIC)
```

Le applicazioni non saranno più in grado di aprire l'argomento per la pubblicazione o effettuare nuove sottoscrizioni utilizzando il nome oggetto, ORANGE.TOPIC. Le applicazioni di pubblicazione che hanno

l'argomento aperto possono continuare a pubblicare la stringa di argomenti risolta. Tutte le sottoscrizioni già effettuate a questo argomento continuano a ricevere le pubblicazioni dopo che l'argomento è stato eliminato.

Le applicazioni che non fanno riferimento a questo oggetto argomento ma che utilizzano la stringa argomento risolta rappresentata da questo oggetto argomento, 'ORANGE' in questo esempio, continuano a funzionare. In questo caso, ereditano le proprietà da un oggetto argomento più in alto nella struttura ad albero degli argomenti. Per ulteriori informazioni, consultare [Alberi degli argomenti](#).

### Riferimenti correlati

[Elimina argomento](#)

## Utilizzo delle sottoscrizioni

Utilizzare i comandi MQSC per gestire le sottoscrizioni.

Le sottoscrizioni possono essere di tre tipi, definiti nell'attributo **SUBTYPE** :

### Amministratore

Definito amministrativamente da un utente.

### Proxy

Una sottoscrizione creata internamente per instradare le pubblicazioni tra i gestori code.

### API

Creato in modo programmatico, ad esempio, utilizzando la chiamata MQSUB MQI.

Consultare [Comandi MQSC](#) per informazioni dettagliate su questi comandi.

### Concetti correlati

[“Definizione di una sottoscrizione di gestione” a pagina 171](#)

Utilizzare il comando MQSC **DEFINE SUB** per creare una sottoscrizione di gestione. È anche possibile utilizzare il valore predefinito definito nella definizione della sottoscrizione locale predefinita. Oppure, è possibile modificare le caratteristiche della sottoscrizione da quelle della sottoscrizione locale predefinita, SYSTEM.DEFAULT.SUB che è stato creato quando il sistema è stato installato.

[“Visualizzazione degli attributi delle sottoscrizioni” a pagina 172](#)

È possibile utilizzare il comando **DISPLAY SUB** per visualizzare gli attributi configurati di qualsiasi sottoscrizione nota al gestore code.

[“Modifica degli attributi della sottoscrizione locale” a pagina 173](#)

È possibile modificare gli attributi della sottoscrizione in due modi, utilizzando il comando **ALTER SUB** o il comando **DEFINE SUB** con l'attributo **REPLACE** .

[“Copia di una definizione di sottoscrizione locale” a pagina 174](#)

È possibile copiare una definizione di sottoscrizione utilizzando l'attributo **LIKE** sul comando **DEFINE** .

[“Eliminazione di una sottoscrizione locale” a pagina 174](#)

È possibile utilizzare il comando MQSC **DELETE SUB** per eliminare una sottoscrizione locale.

## Definizione di una sottoscrizione di gestione

Utilizzare il comando MQSC **DEFINE SUB** per creare una sottoscrizione di gestione. È anche possibile utilizzare il valore predefinito definito nella definizione della sottoscrizione locale predefinita. Oppure, è possibile modificare le caratteristiche della sottoscrizione da quelle della sottoscrizione locale predefinita, SYSTEM.DEFAULT.SUB che è stato creato quando il sistema è stato installato.

Ad esempio, il comando **DEFINE SUB** che segue definisce una sottoscrizione denominata ORANGE con le seguenti caratteristiche:

- Sottoscrizione durevole, che indica che persiste al riavvio del gestore code, con scadenza illimitata.
- Ricevere le pubblicazioni effettuate sulla stringa di argomenti ORANGE, con le priorità dei messaggi impostate dalle applicazioni di pubblicazione.

- Le pubblicazioni distribuite per questa sottoscrizione vengono inviate alla coda locale SUBQ, questa coda deve essere definita prima della definizione della sottoscrizione.

```
DEFINE SUB (ORANGE) +
TOPICSTR (ORANGE) +
DESTCLAS (PROVIDED) +
DEST (SUBQ) +
EXPIRY (UNLIMITED) +
PUBPRTY (ASPUB)
```

#### Nota:

- Il nome della sottoscrizione e della stringa di argomenti non devono corrispondere.
- Tranne che per i valori della destinazione e della stringa argomento, tutti i valori di attributo visualizzati sono i valori predefiniti. Sono mostrati qui solo come un'illustrazione. È possibile ometterli se si è certi che i valori predefiniti sono quelli desiderati o non sono stati modificati. Consultare anche [“Visualizzazione degli attributi delle sottoscrizioni”](#) a pagina 172.
- Se si dispone già di una sottoscrizione locale sullo stesso gestore code con il nome ORANGE, questo comando non riesce. Utilizzare l'attributo **REPLACE** se si desidera sovrascrivere la definizione esistente di una coda, ma consultare anche [“Modifica degli attributi della sottoscrizione locale”](#) a pagina 173.
- Se la coda SUBQ non esiste, questo comando ha esito negativo.

#### Riferimenti correlati

[DEFINE SUB](#)

## Visualizzazione degli attributi delle sottoscrizioni

È possibile utilizzare il comando **DISPLAY SUB** per visualizzare gli attributi configurati di qualsiasi sottoscrizione nota al gestore code.

Ad esempio, utilizzare:

```
DISPLAY SUB(ORANGE)
```

È possibile visualizzare in modo selettivo gli attributi specificandoli singolarmente. Ad esempio:

```
DISPLAY SUB(ORANGE) +
SUBID +
TOPICSTR +
DURABLE
```

Questo comando visualizza i tre attributi specificati nel modo seguente:

```
AMQ8096: IBM MQ subscription inquired.
SUBID(414D51204141412020202020202020EE921E4E20002A03)
SUB(ORANGE)
DURABLE(YES)
TOPICSTR(ORANGE)
```

TOPICSTR è la stringa argomento risolta su cui questo sottoscrittore è operativo. Quando una richiesta viene definita per utilizzare un oggetto argomento, la stringa di argomenti da tale oggetto viene utilizzata come prefisso alla stringa di argomenti fornita durante l'esecuzione della richiesta. SUBID è un identificativo univoco assegnato dal gestore code quando viene creata una sottoscrizione. Questo è un attributo utile da visualizzare perché alcuni nomi di sottoscrizioni potrebbero essere lunghi o in serie di caratteri differenti per i quali potrebbero diventare impraticabili.

Un metodo alternativo per visualizzare le sottoscrizioni consiste nell'utilizzare il SUBID:

```
DISPLAY SUB +
SUBID(414D51204141412020202020202020EE921E4E20002A03) +
TOPICSTR +
DURABLE
```

Questo comando fornisce lo stesso output di prima:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

Le sottoscrizioni proxy su un gestore code non vengono visualizzate per default. Per visualizzarli specificare un **SUBTYPE** di PROXY o ALL.

È possibile utilizzare il comando [DISPLAY SBSTATUS](#) per visualizzare gli attributi di runtime. Ad esempio, utilizzare il comando:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGS
```

Viene visualizzato il seguente output:

```
AMQ8099: IBM MQ subscription status inquired.  
SUB(ORANGE)  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
NUMMSGS(0)
```

Quando si definisce una sottoscrizione di gestione, vengono utilizzati tutti gli attributi non specificati esplicitamente dalla sottoscrizione predefinita, denominata SYSTEM.DEFAULT.SUB. Per visualizzare questi attributi predefiniti, utilizzare il seguente comando:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

### Riferimenti correlati

[VISUALIZZA SECONDARIO](#)

## Modifica degli attributi della sottoscrizione locale

È possibile modificare gli attributi della sottoscrizione in due modi, utilizzando il comando **ALTER SUB** o il comando **DEFINE SUB** con l'attributo **REPLACE**.

Se, ad esempio, si desidera modificare la priorità dei messaggi consegnati a una sottoscrizione denominata ORANGE in modo che sia 5, utilizzare uno dei seguenti comandi:

- Utilizzando il comando **ALTER** :

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Questo comando modifica un singolo attributo, quello della priorità dei messaggi consegnati a questa sottoscrizione in 5; tutti gli altri attributi rimangono gli stessi.

- Utilizzando il comando **DEFINE** :

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

Questo comando modifica non solo la priorità dei messaggi consegnati a questa sottoscrizione, ma anche tutti gli altri attributi ai quali vengono assegnati valori predefiniti.

Se si modifica la priorità dei messaggi inviati a questa sottoscrizione, i messaggi esistenti non vengono influenzati. Tutti i nuovi messaggi, tuttavia, hanno la priorità specificata.

### Riferimenti correlati

[MODIFICA SUB](#)

[DEFINE SUB](#)

## Copia di una definizione di sottoscrizione locale

È possibile copiare una definizione di sottoscrizione utilizzando l'attributo **LIKE** sul comando **DEFINE** .

Ad esempio:

```
DEFINE SUB(BLUE) +  
      LIKE(ORANGE)
```

È anche possibile copiare gli attributi del REAL secondario nel THIRD.SUB e specificare che il correlID delle pubblicazioni fornite è THIRD, piuttosto che il correlID dei publisher. Ad esempio:

```
DEFINE SUB(THIRD.SUB) +  
      LIKE(BLUE) +  
      DESTCORL(ORANGE)
```

### Riferimenti correlati

[DEFINE SUB](#)

## Eliminazione di una sottoscrizione locale

È possibile utilizzare il comando MQSC **DELETE SUB** per eliminare una sottoscrizione locale.

```
DELETE SUB(ORANGE)
```

È anche possibile eliminare una sottoscrizione utilizzando il SUBID:

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

### Riferimenti correlati

[ELIMINA SUB](#)

## Controllo dei messaggi su una sottoscrizione

Quando una sottoscrizione è definita, è associata a una coda. I messaggi pubblicati corrispondenti a questa sottoscrizione vengono inseriti in questa coda.

### Informazioni su questa attività

Tenere presente che i seguenti comandi **runmqsc** mostrano solo le sottoscrizioni che hanno ricevuto i messaggi.

Per controllare i messaggi attualmente accodati per una sottoscrizione, effettuare le seguenti operazioni:

### Procedura

1. Per controllare i messaggi in coda per un tipo sottoscrizione `DISPLAY SBSTATUS(sub_name)` `NUMMSGS`, consultare [“Visualizzazione degli attributi delle sottoscrizioni”](#) a pagina 172.
2. Se il valore di **NUMMSGS** è maggiore di zero, identificare la coda associata alla sottoscrizione immettendo `DISPLAY SUB(sub_name) DEST`.
3. Utilizzando il nome della coda restituita, è possibile visualizzare i messaggi seguendo la tecnica descritta in [“Ricerca delle code con il programma di esempio”](#) a pagina 138.

### Riferimenti correlati

[VISUALIZZAZIONE STATO SB](#)

## Gestione dei servizi

Gli oggetti di servizio sono un mezzo con cui è possibile gestire ulteriori processi come parte di un gestore code. Con i servizi, è possibile definire programmi che vengono avviati e arrestati all'avvio e alla fine del gestore code. I servizi IBM MQ vengono sempre avviati con l'ID utente dell'utente che ha avviato il gestore code.

Per definire una nuova definizione servizio IBM MQ, utilizzare il comando MQSC DEFINE SERVICE.

Gli oggetti di servizio possono essere uno dei seguenti tipi:

### Server

Un server è un oggetto di servizio con il parametro **SERVTYPE** specificato come SERVER. Un oggetto servizio server è la definizione di un programma eseguito quando viene avviato un gestore code specificato. Gli oggetti di servizio del server definiscono i programmi che generalmente vengono eseguiti per un lungo periodo di tempo. Ad esempio, un oggetto di servizio del server può essere utilizzato per eseguire un processo di controllo dei trigger, come **runmqtrm**.

Solo un'istanza di un oggetto servizio server può essere eseguita simultaneamente. Lo stato degli oggetti del servizio server in esecuzione può essere monitorato utilizzando il comando MQSC, **DISPLAY SVSTATUS**.

### Comando

Un comando è un oggetto di servizio che ha il parametro **SERVTYPE** specificato come COMANDO. Gli oggetti servizio comandi sono simili agli oggetti servizio server, tuttavia più istanze di un oggetto servizio comandi possono essere eseguite contemporaneamente e il loro stato non può essere monitorato utilizzando il comando MQSC **DISPLAY SVSTATUS**.

Se il comando MQSC, **STOP SERVICE**, viene eseguito, non viene eseguito alcun controllo per determinare se il programma avviato dal comando MQSC, **START SERVICE**, è ancora attivo prima di arrestare il programma.

### Riferimenti correlati

[VISUALIZZA SVSTATUS](#)

[Avvia servizio](#)

[Arresta servizio](#)

## Definizione di un oggetto servizio

Si definisce un oggetto servizio con il comando MQSC **DEFINE SERVICE**.

Gli attributi da definire sono i seguenti:

### **SERVTYPE**

Definisce il tipo di oggetto servizio. I valori possibili sono i seguenti:

#### **SERVER**

Un oggetto servizio server.

È possibile eseguire una sola istanza di un oggetto servizio server alla volta. Lo stato degli oggetti del servizio server può essere monitorato utilizzando il comando MQSC, **DISPLAY SVSTATUS**.

#### **COMANDO**

Un oggetto servizio comandi.

È possibile eseguire più istanze di un oggetto servizio comandi contemporaneamente. Lo stato di un oggetto servizio comandi non può essere monitorato.

### **STARTCMD**

Il programma eseguito per avviare il servizio. È necessario specificare un percorso completo per il programma.

### **STARTARG**

Argomenti passati al programma di avvio.

## **STDERR**

Specifica il percorso di un file a cui reindirizzare l'errore standard (stderr) del programma di servizio.

## **STDOUT**

Specifica il percorso di un file a cui reindirizzare l'emissione standard (stdout) del programma di servizio.

## **STOPCMD**

Il programma eseguito per arrestare il servizio. È necessario specificare un percorso completo per il programma.

## **STOPARG**

Argomenti passati al programma di arresto.

## **CONTROL**

Specifica il modo in cui il servizio deve essere avviato e arrestato:

### **MANUAL**

Il servizio non deve essere avviato automaticamente o arrestato automaticamente. Viene controllato utilizzando i comandi **START SERVICE** e **STOP SERVICE**. Questo è il valore predefinito.

### **QMGR**

Il servizio da definire deve essere avviato e arrestato contemporaneamente all'avvio e all'arresto del gestore code.

### **SOLO**

Il servizio deve essere avviato contemporaneamente all'avvio del gestore code, ma non viene richiesto l'arresto quando il gestore code viene arrestato.

## **Concetti correlati**

[“Gestione dei servizi” a pagina 176](#)

Utilizzando il parametro **CONTROL**, un'istanza di un oggetto servizio può essere avviata e arrestata automaticamente dal gestore code oppure avviata e arrestata utilizzando i comandi MQSC **START SERVICE** e **STOP SERVICE**.

## **Riferimenti correlati**

[Definisci servizio](#)

[VISUALIZZA SVSTATUS](#)

[Avvia servizio](#)

[Arresta servizio](#)

## **Gestione dei servizi**

Utilizzando il parametro **CONTROL**, un'istanza di un oggetto servizio può essere avviata e arrestata automaticamente dal gestore code oppure avviata e arrestata utilizzando i comandi MQSC **START SERVICE** e **STOP SERVICE**.

Quando viene avviata un'istanza di un oggetto servizio, viene scritto un messaggio nel log degli errori del gestore code contenente il nome dell'oggetto servizio e l'ID processo del processo avviato. Di seguito è riportata una voce di log di esempio per un oggetto di servizio del server in fase di avvio:

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).
```

```
EXPLANATION:
The Server process has started.
ACTION:
None.
```

Di seguito è riportata una voce di log di esempio per un oggetto servizio comandi in fase di avvio:

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
```

```
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5030: The Command 'C1' has started. ProcessId(13030).
```

```
EXPLANATION:
The Command has started.
ACTION:
None.
```

Quando un servizio del server delle istanze si arresta, viene scritto un messaggio nei file di log degli errori del gestore code contenente il nome del servizio e l'ID processo del processo finale. Di seguito è riportata una voce di log di esempio per un oggetto di servizio del server che si arresta:

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5029: The Server 'S1' has ended. ProcessId(13031).
```

```
EXPLANATION:
The Server process has ended.
ACTION:
None.
```

### Riferimenti correlati

[“Variabili di ambiente aggiuntive” a pagina 177](#)

Quando un servizio viene avviato, l'ambiente in cui viene avviato il processo del servizio viene ereditato dall'ambiente del gestore code. È possibile definire ulteriori variabili di ambiente da impostare nell'ambiente del processo del servizio aggiungendo le variabili che si desidera definire a uno dei file di sovrascrittura dell'ambiente `service.env`.

[Arresta servizio](#)

[Avvia servizio](#)

## Variabili di ambiente aggiuntive

Quando un servizio viene avviato, l'ambiente in cui viene avviato il processo del servizio viene ereditato dall'ambiente del gestore code. È possibile definire ulteriori variabili di ambiente da impostare nell'ambiente del processo del servizio aggiungendo le variabili che si desidera definire a uno dei file di sovrascrittura dell'ambiente `service.env`.

## File a cui è possibile aggiungere variabili di ambiente

Esistono due possibili file a cui è possibile aggiungere le variabili di ambiente:

### Il file `service.env` dell'ambito macchina

Questo file si trova in:

-  `/var/mqm` su sistemi AIX and Linux .
-  La directory di dati selezionata durante l'installazione su sistemi Windows .

### Il file `service.env` dell'ambito del gestore code

Questo file si trova nella directory dei dati del gestore code. Ad esempio, l'ubicazione del file di sovrascrittura ambiente per un gestore code denominato QMNAME è:

-  Su sistemi AIX and Linux , `/var/mqm/qmgrs/QMNAME/service.env`
-  Su sistemi Windows , `C:\ProgramData\IBM\MQ\qmgrs\QMNAME\service.env`

Entrambi i file vengono elaborati, se disponibili, con le definizioni nel file di ambito del gestore code che hanno la precedenza su quelle nel file di ambito della macchina.

## Le variabili di ambiente che possono essere specificate in service.env.

Qualsiasi variabile di ambiente può essere specificata in service.env. Ad esempio, se il servizio IBM MQ esegue un numero di comandi, potrebbe essere utile impostare la variabile utente PATH nel file service.env. I valori impostati sulla variabile non possono essere variabili di ambiente; ad esempio CLASSPATH= %CLASSPATH% non è corretto. Allo stesso modo, su Linux PATH= \$PATH : /opt/mqm/bin si ottengono risultati non previsti.

CLASSPATH deve essere maiuscolo e l'istruzione del percorso classe può contenere solo valori letterali. Alcuni servizi (ad esempio Telemetria) impostano il proprio percorso di classe. Il CLASSPATH definito in service.env viene aggiunto ad esso.

Il formato delle variabili definite nel file, service.env è un elenco di coppie di variabili nome e valore. Ogni variabile deve essere definita su una nuova linea e ogni variabile viene presa come è esplicitamente definita, incluso lo spazio.

## Esempio di service.env

```
#####  
##  
## <N_OCO_COPYRIGHT> ##  
## Licensed Materials - Property of IBM ##  
## ##  
## 63H9336 ##  
## (C) Copyright IBM Corporation 2005, 2024. ##  
## ##  
## <NOC_COPYRIGHT> ##  
## ##  
#####  
## ***** ##  
## Module Name: service.env ##  
## Type : IBM MQ service environment file ##  
## Function : Define additional environment variables to be set ##  
## for SERVICE programs. ##  
## Usage : <VARIABLE>=<VALUE> ##  
## ##  
## ***** ##  
MYLOC=/opt/myloc/bin  
MYTMP=/tmp  
TRACEDIR=/tmp/trace  
MYINITQ=ACCOUNTS.INITIATION.QUEUE
```

### Riferimenti correlati

“Inserimenti sostituibili sulle definizioni di servizio” a pagina 178

Nella definizione di un oggetto di servizio, è possibile sostituire i token. I token che vengono sostituiti vengono automaticamente sostituiti con il relativo testo espanso quando viene eseguito il programma di servizio. I token di sostituzione possono essere presi dal seguente elenco di token comuni o da qualsiasi variabile definita nel file, service.env.

## Inserimenti sostituibili sulle definizioni di servizio

Nella definizione di un oggetto di servizio, è possibile sostituire i token. I token che vengono sostituiti vengono automaticamente sostituiti con il relativo testo espanso quando viene eseguito il programma di servizio. I token di sostituzione possono essere presi dal seguente elenco di token comuni o da qualsiasi variabile definita nel file, service.env.

Di seguito sono riportati i token comuni che possono essere utilizzati per sostituire i token nella definizione di un oggetto servizio:

### PERCORSO MQ\_INSTALL\_

L'ubicazione in cui è installato IBM MQ .

### MQ\_DATA\_PATH

L'ubicazione della directory di dati IBM MQ :

-   Su sistemi AIX and Linux , l'ubicazione della directory dei dati IBM MQ è /var/mqm/

- **Windows** Su sistemi Windows , l'ubicazione della directory di dati IBM MQ è la directory di dati selezionata durante l'installazione di IBM MQ

#### **QMNAME**

Il nome del gestore code corrente.

#### **NOME\_SERVIZIO\_MQ**

Il nome del servizio.

#### **PID\_MQ\_SERVER\_**

Questo token può essere utilizzato soltanto dagli argomenti **STOPARG** e **STOPCMD** .

Per gli oggetti servizio server questo token viene sostituito con l'ID processo del processo avviato dagli argomenti **STARTCMD** e **STARTARG** . Altrimenti, questo token viene sostituito con 0.

#### **MQ\_Q\_MGR\_DATA\_PATH**

L'ubicazione della directory dei dati del gestore code.

#### **MQ\_Q\_MGR\_DATA\_NAME**

Il nome trasformato del gestore code. Per ulteriori informazioni sulla trasformazione dei nomi, consultare [Informazioni sui nomi file IBM MQ](#).

Per utilizzare gli inserimenti sostituibili, inserire il token all'interno dei caratteri + in una delle stringhe **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** o **STDERR** . Per esempi, consultare [“Esempi di utilizzo di oggetti di servizio”](#) a pagina 179.

## **Esempi di utilizzo di oggetti di servizio**

I servizi in questa sezione sono scritti con i caratteri di separazione del percorso di stile UNIX , tranne dove diversamente indicato.

### **Utilizzo di un oggetto servizio server**

Questo esempio mostra come definire, utilizzare e modificare un oggetto servizio server per avviare un controllo trigger.

1. Un oggetto servizio server è definito, utilizzando il comando MQSC **DEFINE SERVICE** :

```
DEFINE SERVICE(S1) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+bin/runmqtrm') +
STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +
STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +
STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

Dove:

+MQ\_INSTALL\_PATH+ è un token che rappresenta la directory di installazione.

+QMNAME+ è un token che rappresenta il nome del gestore code.

ACCOUNTS.INITIATION.QUEUE è la coda di avvio.

amqsstop è un programma di esempio fornito con IBM MQ che richiede al gestore code di interrompere tutte le connessioni per l'ID processo. amqsstop genera comandi PCF, quindi il server dei comandi deve essere in esecuzione.

+MQ\_SERVER\_PID+ è un token che rappresenta l'ID del processo passato al programma di arresto.

Consultare [“Inserimenti sostituibili sulle definizioni di servizio”](#) a pagina 178 per un elenco dei token comuni.

2. Un'istanza dell'oggetto del servizio server verrà eseguita al successivo avvio del gestore code. Tuttavia, avvieremo immediatamente un'istanza dell'oggetto servizio server con il comando MQSC **START SERVICE** :

```
START SERVICE(S1)
```

- Viene visualizzato lo stato del processo del servizio server, utilizzando il comando **DISPLAY SVSTATUS MQSC**:

```
DISPLAY SVSTATUS(S1)
```

- Questo esempio mostra ora come modificare l'oggetto servizio del server e come far rilevare gli aggiornamenti riavviando manualmente il processo servizio del server. L'oggetto servizio server viene modificato in modo che la coda di iniziazione sia specificata come JUPITER.INITIATION.QUEUE. Il comando MQSC **ALTER SERVICE** viene utilizzato:

```
ALTER SERVICE(S1) +  
STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

**Nota:** Un servizio in esecuzione non ritira alcun aggiornamento alla propria definizione di servizio fino a quando non viene riavviato.

- Il processo del servizio server viene riavviato in modo che la modifica venga rilevata, utilizzando i comandi **STOP SERVICE** e **START SERVICE MQSC**:

```
STOP SERVICE(S1)
```

seguito da:

```
START SERVICE(S1)
```

Il processo del servizio server viene riavviato e prende le modifiche apportate in “4” a pagina 180.

**Nota:** Il comando MQSC, **STOP SERVICE**, può essere utilizzato solo se nella definizione del servizio è specificato un argomento **STOPCMD**.

### Riferimenti correlati

[MODIFICA SERVIZIO](#)

[Definisci servizio](#)

[VISUALIZZA SVSTATUS](#)

[Avvia servizio](#)

[Arresta servizio](#)

### Utilizzo di un oggetto servizio comandi

Questo esempio mostra come definire un oggetto servizio comandi per avviare un programma che scrive le voci nel log di sistema del sistema operativo quando un gestore code viene avviato o arrestato.

- L'oggetto del servizio comandi è definito utilizzando il comando MQSC **DEFINE SERVICE** :

```
DEFINE SERVICE(S2) +  
CONTROL(QMGR) +  
SERVTYPE(COMMAND) +  
STARTCMD('/usr/bin/logger') +  
STARTARG('Queue manager +QMNAME+ starting') +  
STOPCMD('/usr/bin/logger') +  
STOPARG('Queue manager +QMNAME+ stopping')
```

Dove:

logger è il comando fornito dal sistema AIX o Linux per scrivere nel log di sistema.  
+QMNAME+ è un token che rappresenta il nome del gestore code.

### Riferimenti correlati

[Definisci servizio](#)

## **Utilizzo di un oggetto del servizio comandi solo quando un gestore code termina**

Questo esempio mostra come definire un oggetto servizio comandi per avviare un programma che scrive le voci nel log di sistema del sistema operativo solo quando un gestore code viene arrestato.

1. L'oggetto del servizio comandi è definito utilizzando il comando MQSC **DEFINE SERVICE** :

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

Dove:

logger è un programma di esempio fornito con IBM MQ che può scrivere voci nel log del sistema operativo.

+QMNAME+ è un token che rappresenta il nome del gestore code.

### **Riferimenti correlati**

[Definisci servizio](#)

## **Ulteriori informazioni sul passaggio di argomenti**

Questo esempio mostra come definire un oggetto servizio server per avviare un programma denominato runserv quando viene avviato un gestore code.

Questo esempio viene scritto con i caratteri separatori del percorso di stile Windows .

Uno degli argomenti da passare al programma iniziale è una stringa contenente uno spazio. Questo argomento deve essere passato come una singola stringa. A tal fine, le virgolette doppie vengono utilizzate come mostrato nel seguente comando per definire l'oggetto servizio comando:

1. L'oggetto del servizio server è definito utilizzando il comando MQSC **DEFINE SERVICE** :

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools "') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

```
DEFINE SERVICE(S4) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools "') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

Dove:

+QMNAME+ è un token che rappresenta il nome del gestore code.

"C:\Program Files\Tools \" è una stringa contenente uno spazio, che verrà passato come una singola stringa.

### **Riferimenti correlati**

[Definisci servizio](#)

## **Avvio automatico di un servizio**

Questo esempio mostra come definire un oggetto servizio del server che può essere utilizzato per avviare automaticamente il Controllo trigger all'avvio del gestore code.

1. L'oggetto del servizio server è definito utilizzando il comando MQSC **DEFINE SERVICE** :

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
```

```
STARTCMD('runmqtm') +  
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

Dove:

+QMNAME+ è un token che rappresenta il nome del gestore code.

+IQNAME+ è una variabile di ambiente definita dall'utente in uno dei file service.env che rappresenta il nome della coda di iniziazione.

### Riferimenti correlati

[Definisci servizio](#)

## Gestione degli oggetti per il trigger

IBM MQ consente di avviare automaticamente un'applicazione quando vengono soddisfatte alcune condizioni su una coda. Ad esempio, si potrebbe voler avviare un'applicazione quando il numero di messaggi su una coda raggiunge un numero specificato. Questa funzione è denominata *attivazione*. È necessario definire gli oggetti che supportano il trigger.

Attivazione descritta in dettaglio in [Avvio delle applicazioni IBM MQ utilizzando i trigger](#).

### Definizione di una coda dell'applicazione per il trigger

Una coda di applicazione è una coda locale utilizzata dalle applicazioni per la messaggistica, tramite MQI. L'attivazione richiede la definizione di un certo numero di attributi della coda sulla coda dell'applicazione.

L'attivazione è abilitata dall'attributo **Trigger** (TRIGGER nei comandi MQSC). In questo esempio, un evento trigger deve essere generato quando ci sono 100 messaggi di priorità 5 o superiore sulla coda locale MOTOR.INSURANCE.QUEUE, come segue:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +  
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
MAXMSGL (2000) +  
DEFPSIST (YES) +  
INITQ (MOTOR.INS.INIT.QUEUE) +  
TRIGGER +  
TRIGTYPE (DEPTH) +  
TRIGDPTH (100)+  
TRIGMPRI (5)
```

dove:

#### **QLOCAL (MOTOR.INSURANCE.QUEUE)**

Indica il nome della coda dell'applicazione che si sta definendo.

#### **PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)**

Indica il nome della definizione di processo che definisce l'applicazione che deve essere avviata da un programma di controllo trigger.

#### **MAXMSGL (2000)**

Indica la lunghezza massima dei messaggi sulla coda.

#### **DEFPSIST (YES)**

Specifica che i messaggi su questa coda sono persistenti per impostazione predefinita.

#### **INITQ (MOTOR.INS.INIT.QUEUE)**

Indica il nome della coda di iniziazione su cui il gestore code deve inserire il messaggio trigger.

#### **TRIGGER**

Indica il valore dell'attributo trigger.

#### **TRIGTYPE (DEPTH)**

Specifica che un evento di trigger viene generato quando il numero di messaggi con la priorità richiesta (TRIGMPRI) raggiunge il numero specificato in TRIGDPTH.

#### **TRIGDPTH (100)**

Indica il numero di messaggi richiesti per generare un evento trigger.

## TRIGMPRI (5)

Indica la priorità dei messaggi che devono essere conteggiati dal gestore code per decidere se generare un evento trigger. Vengono contati solo i messaggi con priorità 5 o superiore.

## Definizione di una coda di iniziazione

Quando si verifica un evento trigger, il gestore code inserisce un messaggio trigger sulla coda di avvio specificata nella definizione della coda dell'applicazione. Le code di iniziazione non dispongono di impostazioni speciali, ma è possibile utilizzare la definizione seguente della coda locale MOTOR.INS.INIT.QUEUE per istruzioni:

```
DEFINE QLOCAL(MOTOR.INS.INIT.QUEUE) +
GET (ENABLED) +
NOSHARE +
NOTRIGGER +
MAXMSGL (2000) +
MAXDEPTH (1000)
```

## Definizione di un processo

Utilizzare il comando DEFINE PROCESS per creare una definizione di processo. Una definizione di processo definisce l'applicazione da utilizzare per elaborare i messaggi dalla coda dell'applicazione. La definizione della coda dell'applicazione denomina il processo da utilizzare e quindi associa la coda dell'applicazione all'applicazione da utilizzare per elaborare i relativi messaggi. Questa operazione viene eseguita tramite l'attributo PROCESS sulla coda dell'applicazione MOTOR.INSURANCE.QUEUE. Il seguente comando MQSC definisce il processo richiesto, MOTOR.INSURANCE.QUOTE.PROCESS, identificato in questo esempio:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

Dove:

### **MOTOR.INSURANCE.QUOTE.PROCESS**

È il nome della definizione del processo.

### **DESCR ('Insurance request message processing')**

Descrive il programma applicativo a cui si riferisce questa definizione. Questo testo viene visualizzato quando si utilizza il comando DISPLAY PROCESS. Ciò consente di identificare le operazioni del processo. Se si utilizzano spazi nella stringa, è necessario racchiudere la stringa tra virgolette singole.

### **APPLTYPE (UNIX)**

Indica il tipo di applicazione da avviare.

### **APPLICID ('/u/admin/test/IRMP01')**

È il nome del file eseguibile dell'applicazione, specificato come nome file completo. Nei sistemi Windows, un tipico valore APPLICID è c:\app1\test\irmp01.exe.

### **USERDATA ('open, close, 235')**

Sono dati definiti dall'utente, che possono essere utilizzati dall'applicazione.

## Visualizzazione degli attributi di una definizione di processo

Utilizzare il comando DISPLAY PROCESS per esaminare i risultati della definizione. Ad esempio:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
```

```
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

È inoltre possibile utilizzare il comando MQSC ALTER PROCESS per modificare una definizione del processo esistente e il comando DELETE PROCESS per eliminare una definizione del processo.

## Utilizzo del programma di utilità dmpmqmsg tra due sistemi

Il programma di utilità **dmpmqmsg** (in precedenza *qload*) consente di copiare o spostare il contenuto di una coda o dei relativi messaggi in un file.

### Panoramica

Il file creato con **dmpmqmsg** può essere salvato come richiesto e utilizzato in un secondo momento per ricaricare i messaggi nella coda.

#### Importante:

1. Il file ha un formato specifico compreso dal programma di utilità. Tuttavia, il file è leggibile, in modo da poterlo aggiornare in un editor prima di ricaricarlo. Se si modifica il file, non è necessario modificarne il formato.
2. Da IBM MQ 9.1, il programma di utilità **dmpmqmsg** viene fornito con il fileset di runtime per AIX, Linux, and Windows, quindi è disponibile sia nel client che nel server IBM MQ . Prima di IBM MQ 9.1 il programma di utilità veniva fornito solo con il pacchetto del server.

I possibili utilizzi sono:

- Salvataggio dei messaggi che si trovano in una coda in un file. Probabilmente per scopi di archiviazione e ricaricamento successivo in una coda.
- Ricaricamento di una coda con messaggi precedentemente salvati in un file.
- Rimozione dei vecchi messaggi da una coda.
- 'Riproducendo' i messaggi di test da una posizione memorizzata, anche mantenendo l'ora corretta tra i messaggi, se necessario.



**Attenzione:** SupportPac MO03 ha utilizzato il parametro **-l** per specificare il bind locale o client. **-l** è stato sostituito dal parametro **-c** .

**-P** viene ora utilizzato per le informazioni della codepage invece di **-c**.

Consultare [dmpmqmsg](#) per ulteriori informazioni sul comando e sui parametri disponibili.

## Esempio di utilizzo del programma di utilità dmpmqmsg su Linux, utilizzando una macchina Windows

Si dispone di un gestore code su una macchina Linux che contiene messaggi su una coda (*Q1*) che si desidera spostare in un'altra coda (*Q2*) nello stesso gestore code. Si desidera avviare il programma di utilità **dmpmqmsg** da una macchina Windows .

La coda (*Q1*) contiene quattro messaggi che sono stati aggiunti utilizzando l'applicazione di esempio **amqspmt** (gestore code locale) o **amqspmtc** (gestore code remoto).

Sulla macchina Linux vedi:

```
display q1(Q1) CURDEPTH
      2 : display q1(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Impostare la variabile di ambiente MQSERVER per puntare al gestore code in Linux. Ad esempio:

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

dove *veracruz* è il nome della macchina.

Eeguire l'utilità **dmpmqmsg** per leggere dalla coda, *Q1*, e memorizzare l'output in `c:\temp\mqqload.txt`.

Connettersi come client remoto al gestore code, *QM\_VER*, in esecuzione sull'host Linux e sulla porta stabilita da MQSERVER. Si ottiene la connessione come un client remoto utilizzando l'attributo: `-c`.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqload.txt -c
Read      - Files:    0  Messages:    4  Bytes:      22
Written - Files:    1  Messages:    4  Bytes:      22
```

Il file di output `c:\temp\mqqload.txt` contiene testo, utilizzando un formato che il programma di utilità **dmpmqmsg** comprende.

Sulla macchina Windows , immettere il comando **dmpmqmsg** (utilizzando l'opzione `-o` anziché l'opzione `-i`) per caricare la coda (*Q2*) sulla macchina Linux da un file sulla macchina Windows :

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqload.txt -c
Read      - Files:    1  Messages:    4  Bytes:      22
Written - Files:    0  Messages:    4  Bytes:      22
```

Sulla macchina Linux , notare che ora sono presenti quattro messaggi nella coda che sono stati ripristinati dal file.

```
display ql(Q2) CURDEPTH
      6 : display ql(Q2) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q2)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Sulla macchina Linux ,

Eliminare i messaggi dalla coda originale.

```
clear qllocal(Q1)
      4 : clear qllocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

Confermare che non ci sono più messaggi sulla coda originale:

```
display ql(Q1) CURDEPTH
      5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(0)
```

Consultare [dmpmqmsg](#) per una descrizione del comando e dei relativi parametri.

### Concetti correlati

“Esempi di utilizzo del programma di utilità **dmpmqmsg**” a pagina 186

Semplici modi in cui è possibile utilizzare il programma di utilità **dmpmqmsg** (in precedenza **qload**).

Questo programma di utilità è incorporato nel prodotto da IBM MQ 8.0.

## Esempi di utilizzo del programma di utilità dmpmqmsg

Semplici modi in cui è possibile utilizzare il programma di utilità **dmpmqmsg** (in precedenza **qload**). Questo programma di utilità è incorporato nel prodotto da IBM MQ 8.0.

Precedentemente il programma di utilità **qload** era disponibile come SupportPac MO03.

### Scarica una coda in un file

Utilizzare le seguenti opzioni sulla riga comandi per salvare i messaggi che si trovano in una coda in un file:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

Questo comando acquisisce una copia dei messaggi dalla coda e li salva nel file specificato.

### Scarica una coda in una serie di file

È possibile scaricare una coda in una serie di file utilizzando un carattere `insert` nel nome file. In questa modalità ogni messaggio viene scritto in un nuovo file:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

Questo comando scarica la coda in file, `myfile1`, `myfile2`, `myfile3e` così via.

### Carica una coda da un file

Per ricaricare una coda con i messaggi salvati in [“Scarica una coda in un file”](#) a pagina 186, utilizzare le seguenti opzioni sulla riga comandi:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Questo comando scarica la coda in file, `myfile1`, `myfile2`, `myfile3e` così via.

### Carica una coda da una serie di file

È possibile caricare una coda da una serie di file utilizzando un carattere `insert` nel nome file. In questa modalità ogni messaggio viene scritto in un nuovo file:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Questo comando carica la coda in file, `myfile1`, `myfile2`, `myfile3e` così via.

### Copiare i messaggi da una coda ad un'altra coda

Sostituire il parametro del file in [“Scarica una coda in un file”](#) a pagina 186 con un altro nome coda e utilizzare le seguenti opzioni:

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

Questo comando consente di copiare i messaggi di una coda in un'altra coda.

### Copiare i primi 100 messaggi da una coda all'altra

Utilizzare il comando nell'esempio precedente e aggiungere l'opzione `-r#100` :

```
dmpmqmsg -m QM1 -i Q1 -o Q2 -r#100
```

## Spostare i messaggi da una coda all'altra

Una variazione su “Carica una coda da un file” a pagina 186. Notare la distinzione tra l'utilizzo di **-i** (minuscolo) che sfoglia solo una coda e **-I** (maiuscolo) che deriva in modo distruttivo da una coda:

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

## Spostare i messaggi più vecchi di un giorno da una coda ad un'altra

Questo esempio mostra l'uso della selezione dell'età. È possibile selezionare i messaggi più vecchi, più giovani o che rientrano in un intervallo di età.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

## Visualizza le età dei messaggi attualmente in coda

Utilizzare le opzioni riportate di seguito sulla riga comandi:

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

## Gestione file messaggi

Dopo aver scaricato il messaggio dalla coda, come in “Scarica una coda in un file” a pagina 186, è possibile modificare il file.

Si potrebbe anche voler modificare il formato del file per utilizzare una delle opzioni di visualizzazione non specificate al momento dello scaricamento della coda.

È possibile utilizzare il programma di utilità **dmpmqmsg** per rielaborare il file nel formato richiesto anche dopo lo scaricamento della coda. Utilizzare le opzioni riportate di seguito sulla riga comandi.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

Consultare [dmpmqmsg](#) per una descrizione del comando e dei relativi parametri.

## Utilizzo di oggetti IBM MQ remoti

È possibile gestire oggetti IBM MQ su gestori code remoti utilizzando i comandi MQSC, PCF o amministrative REST API. Prima di poter utilizzare uno di questi metodi, è necessario definire le code di trasmissione e i canali tra il gestore code locale e il gestore code remoto in modo che i comandi possano essere inviati al gestore code remoto e le risposte ricevute dal gestore code locale. In alternativa, è possibile configurare un cluster di gestore code e utilizzare gli stessi metodi di amministrazione remota.

### Informazioni su questa attività

Per preparare i gestori code per la gestione remota, è necessario configurare i seguenti oggetti sul gestore code locale:

- Un listener.
- Una coda di trasmissione che ha il nome del gestore code remoto.
- Un canale mittente con i dettagli di connessione per il gestore code remoto.
- Un canale ricevente con lo stesso nome del canale mittente sul gestore code remoto.

È inoltre necessario configurare i seguenti oggetti sul gestore code remoto:

- Un listener.
- Una coda di trasmissione con il nome del gestore code locale.

- Un canale mittente che dispone dei dettagli di connessione per il gestore code locale.
- Un canale ricevente con lo stesso nome del canale mittente sul gestore code locale.

Per ulteriori informazioni sulla configurazione di questi oggetti, consultare [“Configurazione dei gestori code per la gestione remota”](#) a pagina 188.

In alternativa, è possibile configurare un cluster di gestore code. Un *cluster* è un gruppo di gestori code configurato in modo che i gestori code possano comunicare direttamente tra loro su una singola rete senza complesse definizioni di coda di trasmissione, canale e coda. I cluster possono essere configurati facilmente e in genere contengono gestori code che sono logicamente correlati in qualche modo e devono condividere dati o applicazioni. Anche il cluster più piccolo riduce i costi di amministrazione del sistema.

La creazione di una rete di gestori code in un cluster implica un minor numero di definizioni rispetto alla creazione di un ambiente di accodamento distribuito tradizionale. Con meno definizioni da creare, è possibile impostare o modificare la rete in modo più rapido e semplice e ridurre il rischio di commettere un errore nelle definizioni.

Per impostare un cluster, è necessaria una definizione di mittente del cluster (CLUSDR) e di destinatario del cluster (CLUSRCVR) per ogni gestore code. Non è necessaria alcuna definizione di coda di trasmissione o di coda remota. I principi dell'amministrazione remota sono gli stessi quando vengono utilizzati all'interno di un cluster, ma le definizioni stesse sono notevolmente semplificate.

Per ulteriori informazioni sulla configurazione di un cluster, consultare [Configurazione di un cluster di gestori code](#).

## Procedura

- Per informazioni su come gestire gli oggetti IBM MQ remoti, consultare i seguenti argomenti secondari:
  - [“Configurazione dei gestori code per la gestione remota”](#) a pagina 188
  - [“Gestione del server dei comandi per la gestione remota”](#) a pagina 192
  - [“Immissione di comandi MQSC su un gestore code remoto”](#) a pagina 193
  - [“Conversione dati tra serie di caratteri codificati”](#) a pagina 195

## Configurazione dei gestori code per la gestione remota

È possibile gestire un gestore code remoto da un gestore code locale utilizzando i comandi amministrative REST API, MQSC o PCF. Il gestore code remoto potrebbe trovarsi sullo stesso sistema, in un'installazione differente o su un altro sistema con lo stesso ambiente o su un altro ambiente IBM MQ. Prima di poter gestire in remoto un gestore code da un gestore code locale, è necessario creare un canale mittente e destinatario, un listener e una coda di trasmissione su ciascun gestore code. Questi canali e code consentono di inviare i comandi al gestore code remoto e di ricevere le risposte sul gestore code locale. La procedura per creare queste code e canali è la stessa se si desidera utilizzare i comandi amministrative REST API, MQSC o PCF.

### Prima di iniziare

- La seguente procedura utilizza i gestori code di esempio `source.queue.manager` e `target.queue.manager`. È necessario creare e avviare questi gestori code sul sistema per seguire questi passi oppure sostituire i propri nomi di gestori code nei passi pertinenti.
- La seguente procedura utilizza TCP/IP come tipo di trasporto. È necessario conoscere l'indirizzo IP di entrambi i sistemi per completare questa attività.
- La seguente procedura crea listener che utilizzano le porte di rete 1818 sul sistema locale e 1819 sul sistema remoto. È possibile utilizzare altre porte, ma è necessario sostituire i valori di porta nei passaggi appropriati.
- È necessario eseguire i comandi nella procedura localmente o su una funzione di rete come Telnet.

## Informazioni su questa attività

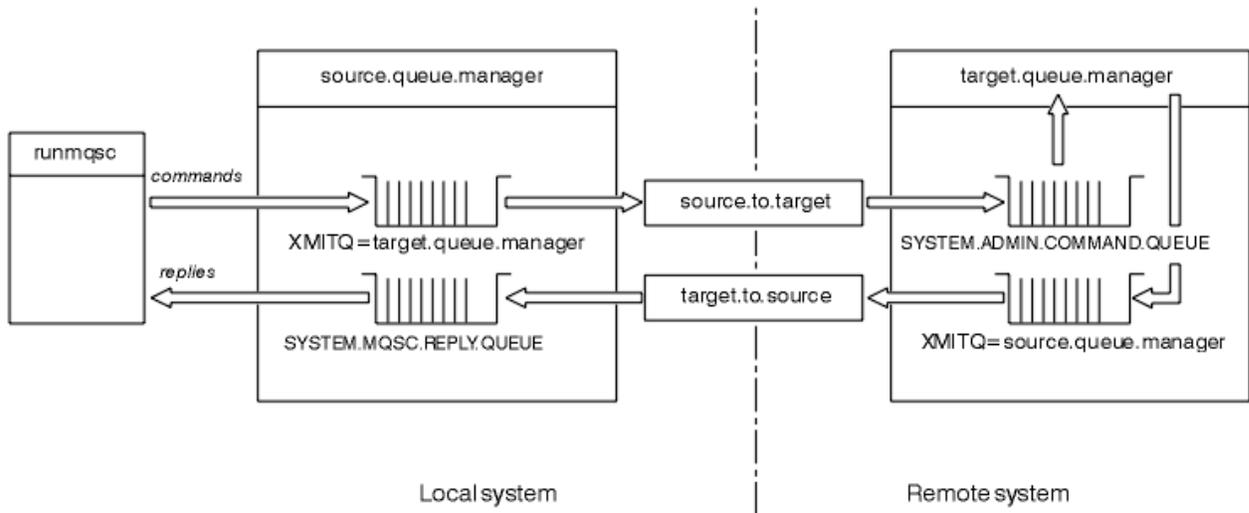


Figura 15. Impostazione di canali e code per l'amministrazione remota

Figura 15 a pagina 189 mostra la configurazione di gestori code, code e canali necessari per la gestione remota:

- L'oggetto `source.queue.manager` è il gestore code di origine da cui è possibile emettere comandi amministrative REST API, MQSC o PCF e a cui vengono restituiti i risultati di questi comandi.
- L'oggetto `target.queue.manager` è il nome del gestore code di destinazione, che elabora i comandi e genera eventuali messaggi dell'operatore.
- I comandi vengono inseriti nella coda di trasmissione con lo stesso nome del gestore code remoto. In questo caso, `target.queue.manager`. Una coda di trasmissione è una coda locale specializzata che conserva temporaneamente i messaggi prima che MCA li raccolga e li invii al gestore code remoto.
- I comandi vengono inviati dal canale `source.to.target` a `SYSTEM.ADMIN.COMMAND.QUEUE` sul gestore code remoto. Ogni estremità del canale ha una definizione separata. Un'estremità è un mittente e l'altra estremità è un destinatario. Le due definizioni devono avere lo stesso nome e insieme costituiscono un singolo canale di messaggi.
- L'output del comando viene inserito nella coda di trasmissione remota con lo stesso nome del gestore code locale da cui è stato inviato il comando. In questo caso, `source.queue.manager`.
- L'output viene inviato dal canale `target.to.source` ad una coda di risposta appropriata, dove viene acquisito ed emesso dal comando originale.

## Procedura

1. Sul gestore code del sistema remoto, verificare la presenza della coda comandi `SYSTEM.ADMIN.COMMAND.QUEUE`. Questa coda viene creata per impostazione predefinita quando viene creato il gestore code.
2. Sul sistema remoto, controllare che il command server sia in esecuzione sul gestore code. Se il server dei comandi non è in esecuzione, la gestione remota non è possibile:
  - a) Avviare **runmqsc** per il gestore code. Ad esempio, per il gestore code `target.queue.manager`, immettere il comando seguente:

```
runmqsc target.queue.manager
```

- b) Visualizzare lo stato del server dei comandi immettendo il seguente comando:

```
DISPLAY QMSTATUS CMDSERV
```

c) Uscire da **runmqsc** immettendo il comando seguente:

```
end
```

d) Se il server dei comandi non è avviato, avviarlo. Ad esempio, per il gestore code `target.queue.manager`, immettere il comando seguente:

```
strmqcsv target.queue.manager
```

3. Definire i canali, il listener e la coda di trasmissione sul gestore code locale:

a) Avviare **runmqsc** per il gestore code. Ad esempio, per il gestore code `source.queue.manager`, immettere il comando seguente:

```
runmqsc source.queue.manager
```

b) Definire il canale mittente. Questo canale mittente deve avere lo stesso nome del canale ricevente sul gestore code remoto. Ad esempio, immettere il seguente comando MQSC, sostituendo il valore per **CONNNAME** con l'indirizzo IP per il gestore code remoto e il numero porta del listener:

```
DEFINE CHANNEL ('source.to.target') +  
CHLTYPE(SDR) +  
CONNNAME (localhost:1819) +  
XMITQ ('target.queue.manager') +  
TRPTYPE(TCP)
```

c) Definire il canale destinatario. Questo canale ricevente deve avere lo stesso nome del canale mittente sul gestore code remoto. Ad esempio, immetti il seguente comando:

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

d) Definire il listener sul gestore code locale. Ad esempio, immetti il seguente comando:

```
DEFINE LISTENER ('source.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1818)
```

e) Definire la coda di trasmissione sul gestore code locale. Questa coda di trasmissione deve avere lo stesso nome del gestore code remoto. Ad esempio, immetti il seguente comando:

```
DEFINE QLOCAL ('target.queue.manager') +  
USAGE (XMITQ)
```

f) Avviare il listener. Ad esempio, immetti il seguente comando:

```
START LISTENER ('source.queue.manager')
```

g) Uscire da **runmqsc** immettendo il comando seguente:

```
end
```

4. Definire i canali, il listener e la coda di trasmissione sul gestore code remoto:

a) Avviare **runmqsc** per il gestore code. Ad esempio, per il gestore code `target.queue.manager`, immettere il comando seguente:

```
runmqsc target.queue.manager
```

b) Definire il canale mittente. Questo canale mittente deve avere lo stesso nome del canale ricevente sul gestore code locale. Ad esempio, immettere il comando MQSC riportato di seguito, sostituendo il valore per **CONNNAME** con l'indirizzo IP per il gestore code locale e il numero di porta per il listener:

```
DEFINE CHANNEL ('target.to.source') +
CHLTYPE(SDR) +
CONNNAME (localhost:1818) +
XMITQ ('source.queue.manager') +
TRPTYPE(TCP)
```

- c) Definire il canale destinatario. Questo canale ricevente deve avere lo stesso nome del canale mittente sul gestore code locale. Ad esempio, immettere il seguente comando:

```
DEFINE CHANNEL ('source.to.target') +
CHLTYPE(RCVR) +
TRPTYPE(TCP)
```

- d) Definire il listener. Ad esempio, immetti il seguente comando:

```
DEFINE LISTENER ('target.queue.manager') +
TRPTYPE (TCP) +
PORT (1819)
```

- e) Definire la coda di trasmissione. Questa coda di trasmissione deve avere lo stesso nome del gestore code locale. Ad esempio, immetti il seguente comando:

```
DEFINE QLOCAL ('source.queue.manager') +
USAGE (XMITQ)
```

- f) Avviare il listener. Ad esempio, immetti il seguente comando:

```
START LISTENER ('target.queue.manager')
```

- g) Uscire da **runmqsc** immettendo il comando seguente:

```
end
```

#### 5. Avviare il canale mittente sul sistema locale:

- a) Avviare **runmqsc** per il gestore code. Ad esempio, per il gestore code `source.queue.manager`, immettere il comando seguente:

```
runmqsc source.queue.manager
```

- b) Avviare il canale mittente. Ad esempio, immetti il seguente comando:

```
START CHANNEL ('source.to.target')
```

- c) Uscire da **runmqsc** immettendo il comando seguente:

```
end
```

#### 6. Avviare il canale mittente sul sistema remoto:

- a) Avviare **runmqsc** per il gestore code. Ad esempio, per il gestore code `target.queue.manager`, immettere il comando seguente:

```
runmqsc target.queue.manager
```

- b) Avviare il canale mittente. Ad esempio, immetti il seguente comando:

```
START CHANNEL ('target.to.source')
```

- c) Uscire da **runmqsc** immettendo il comando seguente:

```
end
```

7. Verificare che la configurazione sia stata completata correttamente inviando un comando MQSC dal sistema locale al gestore code remoto:

- a) Avviare **runmqsc** per il gestore code remoto dal sistema locale. Ad esempio, immetti il seguente comando:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- b) Visualizzare le code sul gestore code remoto immettendo il seguente comando:

```
DISPLAY QUEUE (*)
```

In caso di esito positivo, viene visualizzato un elenco di code dal gestore code remoto.

- c) Se questi passi non funzionano, verificare che i canali su entrambi i sistemi siano in uno stato di esecuzione. Se i canali non sono in esecuzione e non vengono avviati, verificare che i canali e le code di trasmissione siano configurati correttamente e che il server dei comandi sia in esecuzione. Ad esempio, verificare che sia specificato il CONNAME corretto per i canali mittente e che le code di trasmissione abbiano i nomi corretti. Inoltre, controllare i log del gestore code per le eccezioni di sicurezza che potrebbero aiutare a risolvere il problema.

## Risultati

I gestori code sono configurati per gestire in remoto il gestore code remoto dal sistema locale.

## Operazioni successive

- Ulteriori informazioni sulla gestione remota utilizzando i comandi MQSC: [“Immissione di comandi MQSC su un gestore code remoto”](#) a pagina 193
- Ulteriori informazioni sulla scrittura di programmi di gestione utilizzando i comandi PCF: [“Utilizzo di IBM MQ Programmable Command Format”](#) a pagina 25.
- Ulteriori informazioni sull'utilizzo di amministrative REST API per la gestione remota: [“Gestione remota mediante REST API”](#) a pagina 78.

## Gestione del server dei comandi per la gestione remota

A ciascun gestore code è associato un server dei comandi. Un server dei comandi elabora i comandi in entrata dai gestori code remoti o i comandi PCF dalle applicazioni. Presenta i comandi al gestore code per l'elaborazione e restituisce un codice di completamento o un messaggio operatore. È possibile avviare, arrestare e visualizzare lo stato del server dei comandi. Un server dei comandi è obbligatorio per tutte le operazioni di gestione che coinvolgono comandi PCF, MQAI e anche per la gestione remota.

### Prima di iniziare

A seconda del valore dell'attributo del gestore code, **SCMDSERV**, il server dei comandi viene avviato automaticamente all'avvio del gestore code o deve essere avviato manualmente. Se il server dei comandi viene avviato automaticamente, non è possibile utilizzare i comandi strmqcsv o endmqcsv per avviare e arrestare il server dei comandi. È possibile modificare il valore dell'attributo **SCMDSERV** utilizzando il comando MQSC **ALTER QMGR**. Per impostazione predefinita, il server dei comandi viene avviato automaticamente.

L'arresto di un gestore code termina anche il server dei comandi ad esso associato.

### Procedura

- Visualizzare lo stato del server dei comandi:
  - a) Avviare **runmqsc** per il gestore code appropriato immettendo il seguente comando:

```
runmqsc target.queue.manager
```

dove `target.queue.manager` è il gestore code per cui viene visualizzato il server dei comandi.

b) Visualizzare lo stato del server dei comandi immettendo i seguenti comandi MQSC:

```
DISPLAY QMSTATUS CMDSERV
```

c) Uscire da **runmqsc** immettendo il comando seguente:

```
end
```

- Se il server dei comandi non è impostato per l'avvio automatico, avviarlo immettendo il seguente comando:

```
strmqcsv target.queue.manager
```

dove `target.queue.manager` è il gestore code per cui viene avviato il server dei comandi.

- Se il server dei comandi non è impostato per l'avvio automatico, arrestare il server dei comandi immettendo il comando seguente:

```
endmqcsv target.queue.manager
```

dove `target.queue.manager` è il gestore code per cui viene arrestato il server dei comandi.

Per impostazione predefinita, il server dei comandi viene arrestato in modo controllato. È possibile arrestare immediatamente il server dei comandi aggiungendo l'indicatore `-i` al comando.

## Immissione di comandi MQSC su un gestore code remoto

Dopo aver configurato i gestori code per la gestione remota, è possibile utilizzare un particolare formato del comando **runmqsc** su un sistema locale per eseguire i comandi MQSC su un gestore code remoto. Ogni comando viene inviato come PCF Escape alla coda comandi, `SYSTEM.ADMIN.COMMAND.QUEUE`, del gestore code remoto. Le risposte vengono ricevute sul SISTEMA `SYSTEM.MQSC.REPLY.QUEUE`.

### Prima di iniziare

È necessario completare i passaggi in “Configurazione dei gestori code per la gestione remota” a pagina 188 per configurare canali, code di trasmissione, listener e il server dei comandi prima di poter gestire in remoto un gestore code utilizzando i comandi MQSC.

### Procedura

1. Verificare che il server dei comandi sia in esecuzione sul gestore code remoto.

Per informazioni su come avviare il server dei comandi su un gestore code, consultare [“Gestione del server dei comandi per la gestione remota”](#) a pagina 192.

2. Sul gestore code di origine, è possibile quindi eseguire i comandi MQSC in uno dei seguenti modi:

- In modo interattivo, avviando **runmqsc** con i seguenti comandi:

–  Se il gestore code remoto si trova su z/OS, immettere il seguente comando:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

–  Se il gestore code remoto non si trova su z/OS, immettere il seguente comando:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- Da un file di comandi:

a. Inserire i comandi MQSC da eseguire sul sistema remoto in un file di testo, un comando per riga.

- b. Verificare i comandi MQSC sul gestore code locale utilizzando l'indicatore `-v` sul comando **runmqsc** . L'indicatore `-v` verifica la validità dei comandi, ma non li esegue. Tenere presente che alcuni comandi potrebbero avere esito negativo se sono applicabili al gestore code remoto ma non al gestore code locale:

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

`myCmdFile.in` contiene i comandi MQSC da controllare e il file `results.out` contiene i risultati della verifica per i comandi.

- c. Eseguire il file di comandi sul gestore code remoto immettendo uno dei seguenti comandi:

-  Se il gestore code remoto si trova su z/OS, immettere il seguente comando:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in >
results.out
```

-  Se il gestore code remoto non si trova su z/OS, immettere il seguente comando:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in >
results.out
```

I parametri utilizzati sono i seguenti:

**-w secondi**

Specifica che i comandi MQSC vengono eseguiti in modalità indiretta, dove i comandi vengono inseriti nella coda di input del server dei comandi ed eseguiti in ordine.

La variabile *secondi* specifica il periodo di attesa, espresso in secondi, per una risposta dal gestore code remoto. Tutte le risposte ricevute dopo questo orario vengono eliminate, ma i comandi MQSC vengono ancora eseguiti sul gestore code remoto. Il seguente messaggio viene generato sul gestore code locale quando il comando scade:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

Quando si smette di emettere i comandi MQSC, il gestore code locale visualizza tutte le risposte in timeout che sono arrivate ed elimina tutte le altre risposte.

**-x**

Specifica che il gestore code remoto è un gestore code z/OS .

**-m localQMgrNome**

Specifica il nome del gestore code locale che si desidera utilizzare per inoltrare i comandi al gestore code remoto

## Operazioni successive

Se hai difficoltà ad eseguire i comandi MQSC in remoto:

- Verificare che il gestore code remoto sia in esecuzione.
- Controllare che il server dei comandi sia in esecuzione sul sistema remoto.
- Verificare che l'intervallo di disconnessione del canale non sia scaduto. Ad esempio, se un canale è stato avviato ma è stato chiuso dopo un certo periodo di tempo. Questo è particolarmente importante se si avviano i canali manualmente.
- Verificare che le richieste inviate dal gestore code locale abbiano senso al gestore code di destinazione. Ad esempio, le richieste che includono parametri non supportati sul gestore code remoto.
- Consultare anche [Risoluzione dei problemi con i comandi MQSC](#).

## Conversione dati tra serie di caratteri codificati

I dati dei messaggi nei formati definiti da IBM MQ (noti anche come formati integrati) possono essere convertiti dal gestore code da una serie di caratteri codificati a un'altra, a condizione che entrambe le serie di caratteri siano relative a una singola lingua o a un gruppo di lingue simili.

Ad esempio, la conversione tra serie di caratteri codificati con identificativi (CCSID) 850 e 500 è supportata, perché entrambe si applicano alle lingue dell'Europa occidentale.

Per le conversioni di caratteri NL (newline) EBCDIC in ASCII, vedere [Tutti i gestori code](#).

Le conversioni supportate sono definite in [Elaborazione conversione dati](#).

**V 9.2.0** Da IBM MQ 9.2.0, la conversione tra CCSID 37 e 500 è supportata su IBM MQ Appliance, Windows, Linux e macOS.

### Quando un gestore code non può convertire i messaggi in formati integrati

Il gestore code non può convertire automaticamente i messaggi in formati integrati se i relativi CCSID rappresentano gruppi di lingue nazionali differenti. Ad esempio, la conversione tra CCSID 850 e CCSID 1025 (che è una serie di caratteri codificati EBCDIC per le lingue che usano lo script cirillico) non è supportata perché molti dei caratteri in una serie di caratteri codificata non possono essere rappresentati nell'altra. Se si dispone di una rete di gestori code che lavorano in lingue nazionali diverse e la conversione dei dati tra alcune serie di caratteri codificati non è supportata, è possibile abilitare una conversione predefinita.

Per le piattaforme a cui si applica `ccsid_part2.tbl`, consultare [“Specifiche della conversione dati predefinita” a pagina 198](#) utilizzando `ccsid_part2.tbl` per ulteriori informazioni. La conversione dati predefinita su piattaforme diverse da quelle a cui si applica il file `ccsid_part2.tbl` è descritta in [“Conversione dati predefinita” a pagina 196](#).

### Supporto di conversione dati Unicode avanzato

Prima di IBM MQ 9.0, le versioni precedenti del prodotto non supportavano la conversione di dati contenenti punti di codice Unicode oltre il Basic Multilingual Plane (punti di codice sopra U+FFFF). Il supporto per la conversione dei dati Unicode era limitato ai punti di codice definiti nello standard Unicode 3.0, codificati in UTF-8 o UCS-2, un sottoinsieme a larghezza fissa a 2 byte di UTF-16.

Da IBM MQ 9.0, il prodotto supporta tutti i caratteri Unicode definiti nello standard Unicode 8.0 nella conversione dati. Ciò include il supporto completo per UTF-16, incluse le coppie surrogate (una coppia di caratteri UTF-16 a 2 byte nell'intervallo compreso tra X'D800' e X'DFFF' che rappresentano un punto di codice Unicode superiore a U+FFFF).

La combinazione di sequenze di caratteri è supportata anche nei casi in cui un carattere precomposto in un CCSID viene messo in corrispondenza con una sequenza di caratteri di combinazione in un altro CCSID.

La conversione dati da e verso Unicode e CCSID 1388, 1390, 1399, 4933, 5488 e 16884 è stata estesa, su alcune piattaforme, per supportare tutti i punti di codice attualmente definiti per questi CCSID, inclusi quelli che si associano ai punti di codice in piani supplementari Unicode.

Nel caso dei CCSID 1390, 1399 e 16884, sono inclusi i caratteri definiti nello standard JIS X 0213 (JIS2004).

È stato inoltre aggiunto il supporto per la conversione da e verso Unicode e sei nuovi CCSID (da 1374 a 1379).

### File `ccsid_part2.tbl`

Da IBM MQ 9.0 viene fornito un ulteriore file, `ccsid_part2.tbl`.

Il file `ccsid_part2.tbl` ha la precedenza sul file `ccsid.tbl` e:

- Consente di aggiungere o modificare voci CCSID

- Specificare la conversione dati predefinita
- Specificare i dati per i diversi livelli di comando

ccsid\_part2.tbl è applicabile solo alle seguenti piattaforme:

-  Linux - tutte le versioni

-  Windows

 Su IBM MQ for Windows, ccsid\_part2.tbl si trova nella directory *MQDataRoot\conv\table* per default. Inoltre, su IBM MQ for Windows registra tutti i codeset supportati.

 Su IBM MQ for Linux, ccsid\_part2.tbl si trova nella directory *MQDataRoot/conv/table* e i codeset supportati sono contenuti nelle tabelle di conversione fornite da IBM MQ.

Sebbene il file ccsid\_part2.tbl sostituisca il file ccsid.tbl esistente utilizzato nelle precedenti versioni di IBM MQ per fornire ulteriori informazioni CCSID, il file ccsid.tbl continua ad essere analizzato da IBM MQ e pertanto non deve essere eliminato.

Per ulteriori informazioni, consultare [“Il file ccsid\\_part2.tbl” a pagina 197.](#)

## ccsid.tbl file

Su piattaforme diverse da quelle a cui si applica ccsid\_part2.tbl, il file ccsid.tbl viene utilizzato per i seguenti scopi:

-  Su AIX, i codeset supportati vengono mantenuti internamente dal sistema operativo.
- Specifica eventuali codeset aggiuntivi. Per specificare ulteriori codeset, è necessario modificare ccsid.tbl (le istruzioni su come eseguire questa operazione sono fornite nel file).
- Specifica qualsiasi conversione di dati predefinita.

È possibile aggiornare le informazioni registrate in ccsid.tbl; è possibile eseguire questa operazione se, ad esempio, un futuro release del proprio sistema operativo supporta ulteriori serie di caratteri codificati.

## Conversione dati predefinita

Da IBM MQ 9.0, il metodo di conversione dati predefinito è cambiato sulle piattaforme seguenti:

-  Linux - tutte le versioni

-  Windows

Per ulteriori informazioni, consultare [“Specifica della conversione dati predefinita” a pagina 198](#) utilizzo ccsid\_part2.tbl.

Se si impostano canali tra due macchine su cui la conversione dati non è normalmente supportata, è necessario abilitare la conversione dati predefinita per il funzionamento dei canali.

Su piattaforme diverse da quelle a cui si applica ccsid\_part2.tbl, per abilitare la conversione dati predefinita, modificare il file ccsid.tbl per specificare un CCSID EBCDIC predefinito e un CCSID ASCII predefinito. Le istruzioni su come eseguire questa operazione sono incluse nel file. È necessario eseguire questa operazione su tutte le macchine che verranno connesse utilizzando i canali. Riavviare il gestore code per rendere effettive le modifiche.

Il processo di conversione dati predefinito è il seguente:

- Se la conversione tra CCSID di origine e di destinazione non è supportata, ma i CCSID degli ambienti di origine e di destinazione sono entrambi EBCDIC o ASCII, i dati carattere vengono passati all'applicazione di destinazione senza conversione.

- Se un CCSID rappresenta una serie di caratteri codificati ASCII, e l'altro rappresenta una serie di caratteri codificati EBCDIC, IBM MQ converte i dati utilizzando i CCSID di conversione dati predefiniti definiti in `ccsid.tbl`.

**Nota:** Tentare di limitare i caratteri che vengono convertiti a quelli che hanno gli stessi valori di codice nella serie di caratteri codificati specificata per il messaggio e nella serie di caratteri codificati predefinita. Se si utilizza solo la serie di caratteri valida per i nomi degli oggetti IBM MQ (come definito in *Denominazione degli oggetti IBM MQ*) in generale, si soddisferà questo requisito. Le eccezioni si verificano con i CCSID EBCDIC 290, 930, 1279 e 5026 utilizzati in Giappone, dove i caratteri minuscoli hanno codici diversi da quelli utilizzati in altri CCSID EBCDIC.

## Conversione dei messaggi in formati definiti dall'utente

Il gestore code non è in grado di convertire i messaggi in formati definiti dall'utente da una serie di caratteri codificati ad un'altra. Se è necessario convertire i dati in un formato definito dall'utente, è necessario fornire un'uscita di conversione dati per ciascun formato. Non utilizzare CCSID predefiniti per convertire i dati carattere in formati definiti dall'utente. Per ulteriori informazioni sulla conversione dei dati in formati definiti dall'utente e sulla scrittura delle uscite di conversione dati, consultare [Scrittura delle uscite di conversione dati](#).

## Modifica del CCSID del gestore code

Una volta utilizzato il CCSID del comando ALTER QMGR per modificare il CCSID del gestore code, arrestare e riavviare il gestore code per assicurarsi che tutte le applicazioni in esecuzione, inclusi il server dei comandi e i programmi del canale, vengano arrestati e riavviati.

Ciò è necessario poiché tutte le applicazioni in esecuzione quando il CCSID del gestore code viene modificato continuano ad utilizzare il CCSID esistente.

## Il file `ccsid_part2.tbl`

Il file `ccsid_part2.tbl` viene utilizzato per fornire ulteriori informazioni CCSID. Il file `ccsid_part2.tbl` sostituisce il file `ccsid.tbl` utilizzato prima di IBM MQ 9.0.

**Nota:** Il file `ccsid.tbl`, utilizzato prima di IBM MQ 9.0 per fornire ulteriori informazioni CCSID, continua ad essere analizzato da IBM MQ e non deve essere eliminato. Tuttavia, le voci in `ccsid_part2.tbl` hanno la precedenza sulle altre voci in `ccsid.tbl`.

Si consiglia di utilizzare `ccsid_part2.tbl` invece di `ccsid.tbl` perché `ccsid_part2.tbl`:

- Contiene il supporto per i valori di codifica Unicode. Da IBM MQ 9.0, il prodotto supporta tutti i caratteri Unicode definiti nello standard Unicode 8.0 nella conversione dei dati, incluso il supporto completo per UTF-16. Per ulteriori informazioni, consultare [“Conversione dati tra serie di caratteri codificati”](#) a pagina 195.
- Consente di specificare la versione delle voci CCSID, in modo che le voci siano applicabili solo ai livelli di comandi selezionati.

È possibile utilizzare il file `ccsid_part2.tbl` per:

- Aggiungere o modificare voci CCSID
- Specificare la conversione dati predefinita
- Specificare i dati per i diversi livelli di comando

Il file `ccsid_part2.tbl` è applicabile solo alle piattaforme seguenti:

-  Linux - tutte le versioni
-  Windows

L'ubicazione del file `ccsid_part2.tbl` dipende dalla piattaforma:

-  La directory `MQDataRoot/conv/table` su tutte le versioni di Linux.

-  La directory `MQDataRoot\conv\table` su Windows.

## Aggiunta o modifica delle voci CCSID

Una voce del file `ccsid_part2.tbl` ha il formato seguente:

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

Una voce di esempio per CCSID 1200 (UTF-16) è:

```
1200 1200 1200 1200 3 8 0 UTF-16
```

**Nota:** Per ulteriori dettagli sul valore per ACRI, vedere il commento nel file `ccsid_part2.tbl`.

Nel formato `ccsid_part2.tbl`:

### Il tipo può essere:

1=SBCS

2=DBCS

3=MBCS

### La codifica può essere uguale a:

1=EBCDIC

2 = ASCII

3 = ISO

4 = UCS-2

5 = UTF-8

6 = Euc

7 = GB18030

8 = UTF-16

9 = UTF-32

Quando si modifica il file:

- È possibile specificare un commento utilizzando il simbolo `#` all'inizio di una riga. Ciò impedisce a IBM MQ di tentare di analizzare la linea.
- Impossibile fornire commenti in linea.
- Assicurarsi di non creare righe vuote.
- Non aggiungere nuove voci alla fine del file.

Le nuove voci CCSID devono essere aggiunte prima delle informazioni della tabella ACRI.

## Specifiche della conversione dati predefinita

È possibile definire i CCSID di conversione predefiniti, che vengono utilizzati per la conversione tra CCSID ASCII o simili e CCSID EBCDIC, se non è supportata alcuna conversione tra due CCSID.

Se si abilita questa funzione, la conversione predefinita viene utilizzata per la trasmissione e le intestazioni del messaggio e può essere utilizzata anche nella conversione dei dati utente.

Le conversioni predefinite vengono abilitate creando due linee simili alle seguenti:

```
default      0      500      1      1      0  
default      0      850      1      2      0
```

La prima riga imposta il valore predefinito per i CCSID EBCDIC su 500 e la seconda riga imposta il valore predefinito per i CCSID ASCII e simili su 850.

## Specifica di dati per livelli di comando diversi

Per specificare le voci CCSID per i diversi livelli di comandi di IBM MQ , utilizzare un simbolo di due punti seguito dal livello di comando (o livelli di comando) di IBM MQ a cui si desidera applicare la sezione successiva.

Il numero rappresenta il livello di comando minimo in cui deve essere eseguito il gestore code o il client. Ad esempio, se il gestore code corrente è il livello di comando 900 e rileva un indicatore di livello di comando 800 o 900, i CCSID vengono letti.

Tuttavia, un gestore code al livello 800 ignora i CCSID nella sezione 900.

Il livello di comando specificato è applicabile a tutte le voci CCSID incontrate dopo un indicatore del livello di comando, fino a quando non viene trovato un nuovo indicatore del livello di comando.

Se si richiede di impostare il livello di comando su tutti i livelli di comando, specificare il numero zero.

Durante la prima analisi di `ccsid_part2.tbl`, IBM MQ considera tutti i CCSID rilevati come validi per tutti i livelli di comando di IBM MQ.

Il controllo delle versioni inizia ad essere utilizzato solo quando IBM MQ rileva il primo indicatore di livello di comando.

Il seguente frammento di codice mostra un esempio di utilizzo della versione:

```
# Comment Block
# End of Comment Block
# Because no command level flag is specified and we're at the start of the file
# the following CCSIDs will be read on all versions
  819  819    0    819    1    3    0    IS08859-1
  923  923    0    923    1    3    0    IS08859-15
 1051 1051    0   1051    1    3    0    IBM-1051
# The colon :900 below shows that the CCSIDs after will only be for MQ cmd level 900 and above
:900
  8629 437    0    437    1    2    0    IBM-437
 12725 437    0    437    1    2    0    IBM-437
 16821 437    0    437    1    2    0    IBM-437
 20917 437    0    437    1    2    0    IBM-437
# The colon :0 below shows that the CCSIDs after will be for all version of MQ
:0
  4946 850    0    850    1    2    0    IBM-850
 33618 850    0    850    1    2    0    IBM-850
 61697 850    0    850    1    2    0    IBM-850
 61698 850    0    850    1    2    0    IBM-850
```

## Amministrazione Managed File Transfer

Utilizzare i comandi Managed File Transfer per gestire Managed File Transfer. È anche possibile utilizzare IBM MQ Explorer per alcune attività di amministrazione.

### Avvia trasferimento inserendo un messaggio in una coda comandi dell'agent

È anche possibile avviare un trasferimento file inserendo un messaggio di trasferimento file nella coda comandi dell'agent di origine. Un nome coda comandi di esempio è `SYSTEM.FTE.COMMAND.AGENT01`. È necessario assicurarsi che il messaggio raggiunga la coda comandi dell'agent di origine corretto; se il messaggio viene ricevuto da un agent che non corrisponde alle informazioni di origine nell'XML, il messaggio viene rifiutato.

L'XML della richiesta di trasferimento deve essere conforme allo schema `FileTransfer.xsd` e utilizzare l'elemento `<request>` come elemento root. Consultare [Formato del messaggio della richiesta di trasferimento file](#) per informazioni sulla struttura e sul contenuto di un messaggio di richiesta di trasferimento. Il modo in cui si inserisce il messaggio di richiesta di trasferimento su una coda comandi

dell'agente è specifico dell'attività. Ad esempio, è possibile utilizzare l'API IBM MQ Java per inserire un messaggio nella coda in modo programmatico.

## Avvio di un agent MFT

Prima di poter utilizzare un agent Managed File Transfer per un trasferimento file, è necessario avviare l'agent.

### Informazioni su questa attività

È possibile avviare un Managed File Transfer Agent dalla riga comandi. In questo caso, il processo agent si arresta quando si scollega il sistema.

**ALW** Su AIX, Linux, and Windows, è possibile configurare un agent in modo che continui l'esecuzione quando ci si scollega dal sistema e si può continuare a ricevere i trasferimenti file.

**z/OS** Su z/OS, è possibile configurare l'agent in modo che venga avviato come attività avviata da JCL senza la necessità di una sessione interattiva.

Tenere presente che, se un agent rileva un errore irreversibile durante l'esecuzione, viene generata una FDC (first failure data capture) e l'agent viene arrestato.

### Procedura

- Per avviare un agente dalla riga comandi, utilizzare il comando **fteStartAgent** .  
Per ulteriori informazioni, consultare **fteStartAgent**.

- **ALW**

Per configurare un agente in modo che continui l'esecuzione quando ci si scollega dal sistema:

- **Windows** Su Windows, configurare l'agente per l'esecuzione come un servizio Windows . Per ulteriori informazioni, consultare [“Avvio di un agent MFT come servizio Windows” a pagina 200.](#)

- **Linux** **AIX** Su AIX and Linux, configurare l'agent in modo che venga avviato automaticamente durante un riavvio utilizzando un file script. Per ulteriori informazioni, consultare [“Avvio di un agent MFT all'avvio del sistema AIX and Linux” a pagina 202.](#)

- **z/OS**

Su z/OS, configurare l'agent in modo che venga avviato come attività avviata da JCL senza la necessità di una sessione interattiva.

Per ulteriori informazioni, consultare [“Avvio di un agent MFT su z/OS” a pagina 203.](#)

### **Windows** Avvio di un agent MFT come servizio Windows

È possibile avviare un agent come servizio Windows in modo che quando ci si scollega Windows, l'agent continua l'esecuzione e può ricevere trasferimenti file.

### Informazioni su questa attività

Su Windows, quando si avvia un agent dalla riga comandi, il processo agent viene eseguito utilizzando il nome utente utilizzato per accedere a Windows. Quando si scollega il sistema, il processo agent si arresta. Per impedire l'arresto dell'agent, è possibile configurarne l'esecuzione come servizio Windows . L'esecuzione come servizio Windows consente inoltre di configurare gli agent in modo che vengano avviati automaticamente quando l'ambiente Windows viene avviato o riavviato.

Completa la seguente procedura per avviare un agente che viene eseguito come servizio Windows . Per eseguire l'agent come servizio Windows , è necessario che Managed File Transfer sia in esecuzione su una delle versioni Windows supportate. Per un elenco degli ambienti supportati, fare riferimento a [Requisiti di sistema per IBM MQ.](#)

I passi esatti dipendono dal fatto che si sia già creato un agente o che si stia creando un agente. Entrambe le opzioni sono descritte nei seguenti passi.

## Procedura

1. Se si sta creando un agente Managed File Transfer , utilizzare il comando **fteCreateAgent**, **fteCreateCDAgent** o **fteCreateBridgeAgent** . Specificare il parametro **-s** per eseguire l'agente come servizio Windows . Nel seguente esempio, viene creato l'agent AGENT1 , che ha un gestore code agent QMGR1. Il servizio Windows viene eseguito utilizzando un nome utente fteuser, a cui è associata una password ftepassword.

```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftepassword
```

Facoltativamente, è possibile specificare un nome per il servizio dopo il parametro **-s** . Se non si specifica un nome, il servizio viene denominato `mqmftAgentAGENTQMGR`, dove *AGENT* è il nome dell'agent specificato e *QMGR* è il nome del gestore code dell'agent. In questo esempio, il nome predefinito per il servizio è `mqmftAgentAGENT1QMGR1`.

**Nota:** L'account utente Windows specificato utilizzando il parametro **-su** deve disporre dei diritti **Log on as a service** . Per informazioni su come configurarlo, consulta [Guida per l'esecuzione di un agent o logger MFT come servizio Windows](#).

Per ulteriori informazioni, vedi [fteCreateAgent](#), [fteCreateCDAgent: create a Connect:Direct bridge agent](#) o [fteCreateBridgeAgent](#) (crea e configura un MFT agent bridge di protocollo).

2. Se è stato seguito il passo precedente per creare un agent, eseguire i comandi MQSC generati dal comando **fteCreateAgent**, **fteCreateCDAgent** o **fteCreateBridgeAgent** . Questi comandi creano le code IBM MQ necessarie all'agente.

Ad esempio, per un agent denominato *AGENT1*, un gestore code agent denominato *QMGR1* e un gestore code di coordinamento denominato *COORDQMGR1*, eseguire il seguente comando:

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. Se non sono stati seguiti i passi precedenti per creare un agent e si desidera configurare un agent esistente per l'esecuzione come servizio Windows , arrestare prima l'agent, se è in esecuzione, quindi modificarne la configurazione.

a) Il seguente esempio utilizza un agente denominato AGENT1. Esegui il seguente comando:

```
fteStopAgent AGENT1
```

- b) Utilizzare il comando **fteModifyAgent** per configurare l'agente da eseguire come servizio Windows :

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

Per ulteriori informazioni, consultare [fteModifyAgent: eseguire un agent MFT come un Windows servizio](#).

4. Avviare l'agent utilizzando il comando **fteStartAgent** . In alternativa, è possibile utilizzare lo strumento Windows Services, disponibile da Strumenti di amministrazione nel Pannello di controllo, selezionato dal menu di avvio del desktop Windows , per avviare il servizio.

```
fteStartAgent AGENT1
```

Il servizio continua ad essere eseguito anche se ci si scollega da Windows. Per garantire che il servizio venga riavviato anche quando Windows viene riavviato dopo una chiusura, il campo **Tipo di avvio** nello strumento Servizi Windows è impostato su **Automatico** per impostazione predefinita.

Modificare questa impostazione in **Manuale** se non si desidera che il servizio venga riavviato al riavvio di Windows .

5. Opzionale: Per arrestare l'agent, utilizzare il comando `fteStopAgent` oppure utilizzare lo strumento Windows Services. Ad esempio, dalla riga comandi, immetti il seguente comando:

```
fteStopAgent AGENT1
```

- Quando esegui il comando **fteStopAgent** come un servizio, il comando viene sempre eseguito utilizzando il parametro **-i** indipendentemente dal fatto che tu abbia specificato o meno questo parametro. Il parametro **-i** arresta immediatamente l'agent senza completare alcun trasferimento in corso. Ciò è causato da una limitazione del servizio Windows .

## Operazioni successive

Se hai problemi ad avviare il tuo servizio Windows , consulta [Guida per l'esecuzione di un agent o di un programma di registrazione MFT come un servizio Windows](#). Questo argomento descrive anche l'ubicazione dei file di log del servizio Windows .

Linux

AIX

## Avvio di un agent MFT all'avvio del sistema AIX and Linux

Un Managed File Transfer Agent può essere configurato per essere avviato all'avvio del sistema su AIX and Linux. Quando ci si scollega, l'agent continua l'esecuzione e può ricevere trasferimenti file.

Una volta creato e configurato un agent utilizzando uno di questi comandi Managed File Transfer ; **fteCreateAgent**, **fteCreateCDAgent**, o **fteCreateBridgeAgent**, è possibile configurarlo in modo che venga avviato automaticamente durante un riavvio su macchine AIX and Linux utilizzando un file di script che esegue semplicemente il seguente comando:

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

Dove `mq_install_root` è la directory root dell'installazione Managed File Transfer richiesta, il valore predefinito è: `/opt/mqm` e `agent_name` è il nome del Managed File Transfer Agent da avviare. L'utilizzo di questo file script varia a seconda del sistema operativo specifico. Ad esempio, ci sono ulteriori opzioni disponibili in Linux.

## Linux

Linux

Per sistemi Linux esistono diversi modi per avviare le applicazioni durante il processo di avvio del sistema. In generale, considerare la seguente procedura:

1. Creare un file denominato `/etc/rc.mqmft` con contenuto:

```
#!/bin/sh
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

Dove `mqmft_user` è l'ID utente con cui deve essere eseguito il processo dell'agent. Questo ID utente deve essere un membro del gruppo `mqm`.

2. Rendere il file eseguibile, ad esempio:

```
chmod 755 /etc/rc.mqmft
```

3. Aggiungere quindi la riga seguente a `/etc/inittab`:

```
mqmft:5:boot:/etc/rc.mqmft
```

Altri modi per avviare un agent durante l'avvio su Linux includono l'aggiunta di righe di script al file `/etc/rc.d/rc.local` o su Linux SuSe, aggiungendo righe di script al file `/etc/init.d/boot.local`. È necessario selezionare il metodo che funziona meglio per il proprio ambiente. Di seguito sono riportate ulteriori informazioni su altri modi per avviare un agent durante l'avvio su specifiche distribuzioni Linux supportate:

### SLES 10 e 11

Per sistemi SUSE Linux Enterprise Server (SLES) 10 e 11, attenersi alla seguente procedura:

1. Come ID utente root di sistema, creare il file `/etc/init.d/rc.rclocal`.
2. Aggiungere le seguenti righe al file `rc.rclocal`:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

3. Eseguire i seguenti comandi:

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

## Avvio degli agent Managed File Transfer su Linux con systemd

### Linux

Effettuare la seguente procedura:

1. Creare un file nella cartella di sistema `/etc/systemd/system/` e denominarlo, ad esempio `<agentname>.service`. Aggiungere il seguente contenuto, dove `<agentname>` è `MFT_AGT_LNX_0`.

```
# vi /etc/systemd/system/MFT_AGT_LNX_0.service
[Unit]
Description=IBM MQ MFT MFT_AGT_LNX_0
[Service]
ExecStart=/opt/mqm/bin/fteStartAgent MFT_AGT_LNX_0
ExecStop=/opt/mqm/bin/fteStopAgent MFT_AGT_LNX_0
Type=forking
User=mqm
Group=mqm
KillMode=none
```

2. Per abilitare il servizio, eseguire i seguenti comandi:

```
# systemctl enable MFT_AGT_LNX_0
# systemctl daemon-reload
```

3. Per avviare l'agent e controllarne lo stato, eseguire i seguenti comandi:

```
# systemctl start MFT_AGT_LNX_0
# systemctl status MFT_AGT_LNX_0
```

### z/OS Avvio di un agent MFT su z/OS

Su z/OS, oltre ad eseguire il comando `fteStartAgent` da una sessione di z/OS UNIX System Services, è possibile avviare un agent come attività avviata da JCL senza la necessità di una sessione interattiva.

Un'attività avviata viene utilizzata poiché viene eseguita con un ID utente specifico e non è influenzata dalla disconnessione degli utenti.

**Nota:** Le attività avviate vengono generalmente eseguite con un utente di gestione che potrebbe non disporre dei privilegi di accesso e quindi non è possibile accedere al sistema z/OS come l'utente con cui è in esecuzione l'agent. I comandi **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** e il comando **fteShowAgentDetails** con il parametro **-d** specificato non possono essere emessi per tale agent.

Da IBM MQ 9.0.2 e IBM MQ 9.0.0 Fix Pack 1, è possibile utilizzare la proprietà agent **adminGroup** con gli agent Managed File Transfer su z/OS. È possibile definire un gruppo di gestori della sicurezza, ad esempio MFTADMIN, quindi aggiungere l'id utente dell'attività avviata e gli id TSO dell'amministratore a questo gruppo. Modificare il file delle proprietà dell'agente e impostare la proprietà **adminGroup** in modo che sia il nome di questo gruppo del gestore sicurezza.

```
adminGroup=MFTADMIN
```

I membri di questo gruppo possono quindi immettere i comandi **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** e il comando di **fteShowAgentDetails** con il parametro **-d** specificato, per l'agente in esecuzione come attività avviata.

Per ulteriori informazioni, consultare la proprietà **adminGroup** in [Il file MFT agent.properties](#).

Come applicazione Java, un agent è un'applicazione z/OS UNIX System Services che è possibile eseguire da JCL utilizzando il membro BFGAGSTP, da un dataset della libreria PDSE del comando Managed File Transfer generato per un agent. Per ulteriori informazioni su come creare un dataset della libreria PDSE del comando MFT e personalizzarlo per l'agent richiesto, consultare [Creazione di un dataset del comando MFT Agent o Logger](#).

## Abilitazione degli agent per connettersi a gestori code z/OS remoti

LTS

Un agente MFT su z/OS può connettersi a un gestore code su z/OS utilizzando una connessione client nei seguenti scenari:

- L'agent MFT si trova in IBM MQ 9.2.0 Long Term Support, con l'applicazione dell'APAR PH56722 ed è stato associato all'identificativo del prodotto (PID) di IBM MQ Advanced for z/OS Value Unit Editiono IBM MQ Advanced for z/OS
- L'agent MFT si trova in IBM MQ 9.2.0 ed è stato associato al PID di IBM MQ Advanced for z/OS Value Unit Edition

Consultare [IBM MQ ID prodotto e informazioni di esportazione](#) per i dettagli dei prodotti IBM MQ, dei valori PID associati e delle classificazioni di esportazione.

Per informazioni su come impostare il PID associato a un'installazione MFT, consultare **fteSetProductID**.

Il PID con cui è in esecuzione l'agente viene visualizzato nel log all'avvio dell'agente.

**Nota:** Un agent MFT su z/OS, in esecuzione in un qualsiasi altro PID, può connettersi a un gestore code locale solo utilizzando una connessione in modalità bind.

Se un agent tenta di connettersi a un gestore code che non è in esecuzione su z/OS, viene emesso il messaggio BFGQM1044E e l'avvio dell'agent viene terminato.

### Riferimenti correlati

[“Arresto di un agent MFT su z/OS” a pagina 205](#)

Se si sta eseguendo un Managed File Transfer Agent su z/OS come attività avviata da JCL, l'agent accetta i z/OS comandi dell'operatore **MODIFY** e **STOP**, oltre al comando **fteStopAgent**.

[Il file MFT agent.properties](#)

## Elenco di agenti di MFT

È possibile elencare gli agenti Managed File Transfer registrati con un particolare gestore code utilizzando la riga comandi o il IBM MQ Explorer.

### Informazioni su questa attività

Per elencare gli agent utilizzando la riga comandi, consultare [fteListComando degli agent](#).

Per elencare gli agent che utilizzano IBM MQ Explorer, nella vista Navigator fare clic su **Agent** sotto il nome del gestore code di coordinamento.

Se un agent non è elencato dal comando **fteListAgents** o non viene visualizzato in IBM MQ Explorer, utilizzare il diagramma di flusso di diagnosi nel seguente argomento per individuare e correggere il problema: [Cosa fare se l'agent MFT non viene elencato dal comando \*\*fteListAgents\*\*](#).

## Arresto di un agent MFT

È possibile arrestare un agent Managed File Transfer dalla riga comandi. Quando si arresta un agent, si sta disattivando l'agent e si consente all'agent di completare il trasferimento file corrente prima dell'arresto. È inoltre possibile specificare il parametro **-i** sulla riga comandi per arrestare immediatamente un agente. Quando l'agent è stato arrestato, non è possibile utilizzare tale agent per trasferire i file fino a quando non viene riavviato.

### Prima di iniziare

Se si desidera controllare i nomi degli agenti associati ad un gestore code, è possibile elencare gli agenti utilizzando IBM MQ Explorer o la riga comandi, consultare [Comando degli agentiftelist](#).

### Informazioni su questa attività

Per arrestare un agent dalla riga comandi, consultare [fteStopAgent](#).

Se il proprio agent è stato configurato per essere eseguito come servizio Windows, l'esecuzione del comando **fteStopAgent** arresta anche il servizio Windows. In alternativa, è possibile interrompere l'agent arrestando il servizio utilizzando il tool Windows Services. Per ulteriori informazioni, consultare l'argomento [“Avvio di un agent MFT come servizio Windows”](#) a pagina 200.

## Arresto di un agent MFT su z/OS

Se si sta eseguendo un Managed File Transfer Agent su z/OS come attività avviata da JCL, l'agent accetta i z/OS comandi dell'operatore **MODIFY** e **STOP**, oltre al comando **fteStopAgent**.

Un'attività avviata viene utilizzata poiché viene eseguita con un ID utente specifico e non è influenzata dalla disconnessione degli utenti.

**Nota:** Le attività avviate vengono generalmente eseguite con un utente di gestione che potrebbe non disporre dei privilegi di accesso e quindi non è possibile accedere al sistema z/OS come l'utente con cui è in esecuzione l'agent. I comandi **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** e il comando **fteShowAgentDetails** con il parametro **-d** specificato non possono essere emessi per tale agent.

Da IBM MQ 9.0.2 e IBM MQ 9.0.0 Fix Pack 1, è possibile utilizzare la proprietà agent **adminGroup** con gli agent Managed File Transfer su z/OS. È possibile definire un gruppo di gestori della sicurezza, ad esempio MFTADMIN, quindi aggiungere l'id utente dell'attività avviata e gli id TSO dell'amministratore a questo gruppo. Modificare il file delle proprietà dell'agente e impostare la proprietà **adminGroup** in modo che sia il nome di questo gruppo del gestore sicurezza.

```
adminGroup=MFTADMIN
```

I membri di questo gruppo possono quindi immettere i comandi **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** e il comando di **fteShowAgentDetails** con il parametro **-d** specificato, per l'agente in esecuzione come attività avviata.

Per ulteriori informazioni, consultare la proprietà **adminGroup** in [Il file MFT agent.properties](#).

## Arresto controllato dell'agent utilizzando il comando z/OS MODIFY (F)

Il comando **MODIFY** permette di arrestare un agente in modo controllato come alternativa al comando **fteStopAgent**. L'agent completa tutti i trasferimenti attualmente in corso, ma l'agent non avvia nuovi trasferimenti.

Ad esempio:

```
F job_name,APPL=STOP
```

dove *nome\_lavoro* è il lavoro sotto il quale è in esecuzione il processo agent.

## Arresto immediato dell'agent utilizzando il comando z/OS STOP (P)

Il comando **STOP** equivale a un arresto immediato utilizzando il comando **fteStopAgent** con il parametro **-i**. L'agent viene arrestato immediatamente anche se l'agent sta attualmente trasferendo un file.

Ad esempio:

```
P job_name
```

dove *nome\_lavoro* è il lavoro sotto il quale è in esecuzione il processo agent.

### Riferimenti correlati

[“Avvio di un agent MFT su z/OS” a pagina 203](#)

Su z/OS, oltre ad eseguire il comando **fteStartAgent** da una sessione di z/OS UNIX System Services, è possibile avviare un agent come attività avviata da JCL senza la necessità di una sessione interattiva.

[Il file MFT agent.properties](#)

## Avvio di un nuovo trasferimento file

È possibile avviare un nuovo trasferimento file da IBM MQ Explorer o dalla linea di comando ed è possibile scegliere di trasferire un singolo file o più file in un gruppo.

### Informazioni su questa attività

Per avviare un nuovo trasferimento file dalla riga comandi, consultare [fteCreateComando di trasferimento](#).

Per iniziare un nuovo trasferimento file utilizzando il wizard **Crea nuovo trasferimento file gestito** in IBM MQ Explorer, effettuare le seguenti operazioni:

### Procedura

1. Nella vista Navigator, fare clic su **Managed File Transfer. Managed File Transfer Central** viene visualizzato nella vista Contenuto.
2. Tutti i gestori code di coordinamento vengono visualizzati nella vista Navigator. Espandere il nome del gestore code di coordinamento rispetto al quale è registrato l'agente che si desidera utilizzare per il trasferimento. Se al momento si è connessi a un gestore code di coordinamento diverso da quello che si desidera utilizzare per il trasferimento, fare clic con il tasto destro del mouse sul nome del gestore code di coordinamento nella vista Navigator e fare clic su **Disconnetti**. Fare quindi clic con il tasto destro del mouse sul nome del gestore code di coordinamento che si desidera utilizzare e fare clic su **Connetti**.

3. Avviare la procedura guidata **Crea nuovo trasferimento file gestito** utilizzando uno dei metodi seguenti:
  - a) Fare clic con il tasto destro del mouse sul nome di uno dei seguenti nodi nella vista Navigator : il gestore code di coordinamento pertinente, **Modelli di trasferimento**, **Log di trasferimento** o **Trasferimenti in sospeso**. Quindi fare clic su **Nuovo trasferimento** per avviare la procedura guidata.
  - b) Fare clic su **File > Nuovo > Altro > Managed File Transfer Wizards > Procedura guidata Nuovo trasferimento**
4. Seguire le istruzioni sui pannelli della procedura guidata. È inoltre disponibile una guida sensibile al contesto per ciascun pannello. Per accedere alla guida sensibile al contesto su Windows, premere F1. Su Linux, premere Ctrl+F1 o Shift+F1.

## Utilizzo dei file di definizione del trasferimento

È possibile specificare un file di definizione trasferimento che può essere utilizzato per creare un trasferimento file. Il file di definizione del trasferimento è un file XML che definisce alcune o tutte le informazioni richieste per creare il trasferimento.

I file di definizione trasferimento sono utili quando si desidera specificare più file di origine e più file di destinazione in una singola operazione di trasferimento. È possibile utilizzare un file di definizione trasferimento per inoltrare un trasferimento file complesso. È possibile riutilizzare e condividere il file di definizione trasferimento.

È possibile utilizzare due formati per un file di definizione del trasferimento e, mentre questi formati variano leggermente, entrambi sono conformi allo schema `FileTransfer.xsd`. È possibile trovare questo schema nella directory `samples\schema` dell'installazione di Managed File Transfer.

Sono supportati i seguenti due formati di file di definizione trasferimento:

- Una definizione dei file di origine e di destinazione per un trasferimento. Questa definizione utilizza un elemento **transferSpecifications** come root.
- Una definizione dell'intero trasferimento, inclusi i file di origine e di destinazione e gli agenti di origine e di destinazione. Questa definizione utilizza un elemento **request** come root.
  - I file con questo formato possono essere generati dal comando **fteCreateTransfer** utilizzando il parametro **-gt**.

Il seguente esempio mostra un formato file di definizione trasferimento che specifica solo i file di origine e di destinazione per un trasferimento:

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <item checksumMethod="MD5" mode="text">
    <source recursive="false" disposition="leave">
      <file>textTransferTest.txt</file>
    </source>
    <destination type="directory" exist="overwrite">
      <file>c:\targetfiles</file>
    </destination>
  </item>
</transferSpecifications>
```

Per inoltrare questo formato del file di definizione del trasferimento, è necessario specificare gli agenti di origine e di destinazione sulla riga comandi:

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

Il seguente esempio è un formato file di definizione trasferimento che specifica tutte le informazioni richieste per un trasferimento:

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
```

```

xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
<managedTransfer>
  <originator>
    <hostName>example.com.</hostName>
    <userID>fteuser</userID>
  </originator>
  <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
  <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
  <transferSet>
    <item mode="binary" checksumMethod="MD5">
      <source recursive="false" disposition="leave">
        <file>c:\sourcefiles\*.jpg</file>
      </source>
      <destination type="directory" exist="error">
        <file>/targetfiles/images</file>
      </destination>
    </item>
  </transferSet>
</managedTransfer>
</request>

```

È possibile creare un file con questo formato utilizzando il parametro **-gt** sul comando **fteCreateTransfer**. Quando si inoltra un file di definizione trasferimento con questo formato, non è necessario specificare altro sulla riga comandi:

```
fteCreateTransfer -td c:\definitions\example2.xml
```

È possibile sovrascrivere le informazioni sull'agente di origine e di destinazione relative alla riga comandi passando i parametri normali in aggiunta al file di definizione trasferimento. Ad esempio:

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

Questo esempio utilizza le opzioni della riga comandi per sovrascrivere l'agent di destinazione definito all'interno del file di definizione del trasferimento con **AGENT9** e il gestore code di destinazione definito nel file di definizione del trasferimento come **agent9qm**.

Entrambi i formati descritti possono contenere uno o più elementi < item>. Per ulteriori informazioni sull'elemento < item>, consultare [Formato del messaggio di richiesta di trasferimento file](#). Ciascuno di questi elementi di trasferimento definisce una coppia di file di origine e di destinazione con ulteriori attributi per controllare il comportamento del trasferimento. Ad esempio, è possibile specificare il seguente comportamento:

- Se il trasferimento utilizza un checksum
- Se il trasferimento è di testo o binario
- Indica se eliminare il file di origine dopo il completamento del trasferimento
- Indica se sovrascrivere il file di destinazione se il file esiste

Un vantaggio dell'utilizzo dei file delle definizioni di trasferimento è che è possibile specificare opzioni aggiuntive che non sono disponibili dalla riga comandi. Ad esempio, quando si eseguono trasferimenti da messaggio a file, è possibile specificare l'attributo groupId utilizzando un file di definizione trasferimento. Questo attributo specifica l'ID gruppo IBM MQ dei messaggi letti dalla coda. Un altro vantaggio dei file di definizione del trasferimento è che è possibile specificare opzioni differenti per ogni coppia di file. Ad esempio, è possibile specificare se viene utilizzato un checksum o se il file viene trasferito in modalità testo o binaria, file per file. Se si utilizza la riga comandi, si applicano le stesse opzioni per ogni file in un trasferimento.

Ad esempio:

```

<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination1.doc</file>
  </destination>

```

```

</item>
<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>

```

**z/OS** È possibile utilizzare gli elementi per trasferire un file da un sistema distribuito a un sistema z/OS :

```

z/OS
<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>

```

**z/OS** Questo esempio trasferisce il file textTransferTest.txt dall'agent di origine al dataset //TEXT.TRANS.TEST sull'agent di destinazione in modalità testo. Questo trasferimento converte i dati di origine dalla codifica predefinita dell'agent di origine (non è specificato alcun attributo di codifica di origine) nella codepage: IBM-1047.

## Creazione di un trasferimento file pianificato

È possibile pianificare un nuovo trasferimento file da IBM MQ Explorero dalla riga comandi. Il trasferimento pianificato può contenere singoli file o più file in un gruppo. È possibile eseguire un trasferimento file pianificato una sola volta o ripetere il trasferimento più volte.

### Informazioni su questa attività

È possibile impostare una pianificazione di trasferimento file che si verifichi una sola volta o che si verifichi ai seguenti intervalli:

- Ogni minuto
- Orario
- Giornaliero
- Settimanale
- Mensile
- Ogni anno

È possibile quindi specificare le ricorrenze da arrestare nei seguenti punti:

- In una data e ora definite
- Dopo un numero definito di ricorrenze

In alternativa, è possibile specificare che le ricorrenze continuino per sempre.

Se un trasferimento pianificato viene eseguito ogni giorno alla stessa ora, utilizzare l'attributo **adjustScheduleTimeForDaylightSaving** nel file delle proprietà dell'agent per modificare l'ora in

cui viene eseguita la pianificazione quando cambiano gli orologi. Per ulteriori informazioni, consultare [Il file MFT agent.properties](#).

Per creare un nuovo trasferimento file pianificato utilizzando la riga comandi, utilizzare i parametri di pianificazione (**-tb, -ss, -oi, -of, -oce -es**) per il comando `fteCreateTransfer`.

Per creare un nuovo trasferimento file pianificato utilizzando la procedura guidata **Crea nuovo trasferimento file gestito** in IBM MQ Explorer, effettuare le seguenti operazioni:

## Procedura

1. Nella vista Navigator, fare clic su **Managed File Transfer. Managed File Transfer Central** viene visualizzato nella vista Contenuto.
2. Tutti i gestori code di coordinamento vengono visualizzati nella vista Navigator. Espandere il nome del gestore code di coordinamento rispetto al quale è registrato l'agente che si desidera utilizzare per il trasferimento. Se al momento si è connessi a un gestore code di coordinamento diverso da quello che si desidera utilizzare per il trasferimento, fare clic con il tasto destro del mouse sul nome del gestore code di coordinamento nella vista Navigator e fare clic su **Disconnetti**. Fare quindi clic con il tasto destro del mouse sul nome del gestore code di coordinamento che si desidera utilizzare e fare clic su **Connetti**.
3. Avviare la procedura guidata **Crea nuovo trasferimento file gestito** utilizzando i seguenti metodi:
  - a) Fare clic con il tasto destro del mouse sul nome di uno dei seguenti nodi nella vista Navigator: il gestore code di coordinamento pertinente, **Modelli di trasferimento, Log di trasferimentoo Trasferimenti in sospeso**. Quindi fare clic su **Nuovo trasferimento** per avviare la procedura guidata.
  - b) Fare clic su **File > Nuovo > Altro > Managed File Transfer Wizards > Procedura guidata Nuovo trasferimento**
4. Seguire le istruzioni sui pannelli della procedura guidata. Assicurarsi di selezionare la check box **Abilita trasferimento pianificazione** e immettere i dettagli della pianificazione nella scheda **Pianificazione**. I trasferimenti file pianificati iniziano entro un minuto dall'ora di inizio della pianificazione, se non vi sono problemi che potrebbero influire sul trasferimento. Ad esempio, potrebbero verificarsi problemi con la rete o l'agente che impediscono l'avvio del trasferimento pianificato. Per ogni pannello viene fornita una guida sensibile al contesto. Per accedere alla guida sensibile al contesto su Windows, premere F1. Su Linux, premere Ctrl+F1 o Shift+F1.

## Risultati

Per informazioni sui messaggi coinvolti nei trasferimenti file pianificati, consultare [Formati dei messaggi di log dei trasferimenti file pianificati](#).

## Gestione trasferimenti file in sospeso

È possibile visualizzare i trasferimenti file pianificati in sospeso da IBM MQ Explorer. La finestra **Trasferimenti in sospeso** visualizza tutti i trasferimenti in sospeso registrati con il gestore code di coordinamento a cui si è attualmente connessi.

### Informazioni su questa attività

Per visualizzare lo stato di un trasferimento file pianificato non ancora avviato, utilizzare la seguente procedura:

## Procedura

1. Espandere **Managed File Transfer** nella vista Navigator. **Managed File Transfer Central** viene visualizzato nella vista Contenuto.
2. Tutti i gestori code di coordinamento vengono visualizzati nella vista Navigator. Espandere il nome del gestore code di coordinamento utilizzato per il trasferimento pianificato. Se si desidera modificare il

- gestore code di coordinamento a cui si è connessi, fare clic con il tasto destro del mouse sul nome del gestore code di coordinamento che si desidera utilizzare nella vista Navigator e fare clic su **Connetti**.
3. Fare clic su **Trasferimenti in sospeso**. Viene visualizzata la finestra **Trasferimenti in sospeso** nella vista Contenuto.
  4. La finestra **Trasferimenti in sospeso** visualizza i seguenti dettagli sui trasferimenti file pianificati:
    - a) **Nome** Il numero del trasferimento file pianificato. Questo numero viene assegnato automaticamente.
    - b) **Origine** Il nome dell'agent di origine.
    - c) **File di origine** Il nome del file da trasferire sul sistema host.
    - d) **Destinazione** Il nome dell'agent di destinazione.
    - e) **File di destinazione** Il nome del file dopo che è stato trasferito al sistema di destinazione.
    - f) **Avvio pianificato (fuso orario selezionato)** L'ora e la data in cui è pianificato l'avvio del trasferimento file nel fuso orario selezionato dall'amministratore. Per modificare il fuso orario visualizzato, fare clic su **Finestra > Preferenze > IBM MQ Explorer > Managed File Transfer** e selezionare un fuso orario alternativo dall'elenco **Fuso orario:** . Fare clic su **OK**.
    - g) **Ripeti ogni** Se si è scelto di ripetere il trasferimento pianificato, l'intervallo specificato in cui si desidera ripetere il trasferimento, espresso come numero.
    - h) **Tipo di ripetizione** Se si è scelto di ripetere il trasferimento pianificato, il tipo di intervallo di ripetizione specificato per il trasferimento file. Il tipo può essere uno dei seguenti valori: minuti, ore, giorni, settimane, mesio anni.
    - i) **Ripeti fino a** Se si è scelto di ripetere il trasferimento pianificato, i dettagli di quando si desidera arrestare il trasferimento file ripetuto. Ad esempio, una data e ora specificate o dopo un numero specificato di ricorrenze.

## Risultati

Per aggiornare quanto visualizzato nella finestra **Trasferimenti in sospeso** , fare clic sul pulsante **Aggiorna**  sulla barra degli strumenti della vista Contenuto.

Per annullare un trasferimento file in sospeso, fare clic con il pulsante destro del mouse sul trasferimento particolare e fare clic su **Annulla**. L'annullamento di un trasferimento elimina completamente la richiesta di trasferimento file.

## Attivazione di un trasferimento file

È possibile impostare determinate condizioni di trigger su un trasferimento file che deve essere true prima che tale trasferimento possa essere effettuato. Se le condizioni di attivazione non sono true, il trasferimento file non viene eseguito e un messaggio di log viene facoltativamente inoltrato per registrare il fatto che il trasferimento non si è verificato. La richiesta di trasferimento file viene quindi eliminata. Ad esempio, è possibile impostare un trasferimento file che si verifica solo se un file denominato sul sistema in cui si trova l'agente di origine ha una dimensione superiore a quella specificata o se un particolare file denominato esiste sul sistema in cui si trova l'agente di origine. È possibile impostare un trasferimento file attivato da IBM MQ Explorer o dalla riga comandi.

### Informazioni su questa attività

È possibile monitorare continuamente una risorsa per soddisfare una condizione di trigger. Per ulteriori informazioni sul monitoraggio delle risorse, consultare: [“Monitoraggio delle risorse MFT” a pagina 217.](#)

Esistono tre diverse condizioni di attivazione che è possibile impostare. Le condizioni sono le seguenti:

- Se un particolare file esiste sullo stesso sistema dell'agent di origine
- Se un particolare file non esiste sullo stesso sistema dell'agent di origine

- Se un particolare file supera una certa dimensione sul sistema in cui si trova l'agente di origine (la dimensione può essere espressa in byte, KB, MB o GB). Queste unità di misura utilizzano la convezione  $2^{10}$ , ad esempio 1 KB equivale a 1024 byte e 1 MB equivale a 1024 KB.

I tipi di attivazione nell'elenco precedente possono essere combinati in due modi:

- Per una singola condizione, è possibile specificare più di un file sul sistema in cui si trova l'agente di origine. Ciò attiva il trasferimento se uno qualsiasi dei file specificati soddisfa la condizione (operatore booleano OR).
- È possibile specificare più condizioni. Ciò attiva il trasferimento solo se tutte le condizioni sono soddisfatte (operatore booleano AND).

Puoi anche combinare un trasferimento attivato con un trasferimento pianificato. Per ulteriori informazioni, consultare [Creazione di un trasferimento file pianificato](#). In questo caso, le condizioni di trigger vengono valutate nel momento in cui la pianificazione deve essere avviata o per una pianificazione ripetuta ogni volta che la pianificazione deve essere avviata.

I trasferimenti attivati non sono supportati sugli agent bridge di protocollo.

Per creare un trasferimento file attivato utilizzando la riga comandi, utilizzare il parametro **-tr** sul comando `fteCreateTransfer`.

Per creare un trasferimento file pianificato utilizzando la procedura guidata **Crea nuovo trasferimento file gestito** in IBM MQ Explorer, attenersi alla seguente procedura:

## Procedura

1. Nella vista Navigator, fare clic su **Managed File Transfer**. **Managed File Transfer Central** viene visualizzato nella vista Contenuto.
2. Tutti i gestori code di coordinamento vengono visualizzati nella vista Navigator. Espandere il nome del gestore code di coordinamento utilizzato per il trasferimento pianificato. Se si desidera modificare il gestore code di coordinamento a cui si è connessi, fare clic con il tasto destro del mouse sul nome del gestore code di coordinamento che si desidera utilizzare nella vista Navigator e fare clic su **Connetti**.
3. Avviare la procedura guidata **Crea nuovo trasferimento file gestito** utilizzando uno dei metodi seguenti:
  - a) Fare clic con il tasto destro del mouse sul nome di uno dei seguenti nodi nella vista Navigator: il gestore code di coordinamento pertinente, **Modelli di trasferimento**, **Log di trasferimento** o **Trasferimenti in sospenso**. Quindi fare clic su **Nuovo trasferimento** per aprire la procedura guidata.
  - b) Fare clic su **File > Nuovo > Altro > Managed File Transfer Wizards > Procedura guidata Nuovo trasferimento**
4. Seguire le istruzioni sui pannelli della procedura guidata. Assicurarsi di selezionare la casella di spunta **Abilita trasferimento attivato** nella scheda **Trigger** e completare i campi in tale scheda per impostare il trigger. Per ogni pannello viene fornita una guida sensibile al contesto. Per accedere alla guida sensibile al contesto su Windows, premere F1. Su Linux, premere **Ctrl+F1** o **Shift+F1**.

## Monitoraggio dei trasferimenti file in corso

È possibile monitorare un trasferimento file in corso utilizzando la scheda **Managed File Transfer - Avanzamento trasferimento corrente** in IBM MQ Explorer. Questo trasferimento file può essere avviato da IBM MQ Explorer o dalla riga comandi. La scheda visualizza anche l'avanzamento dei trasferimenti pianificati nel momento in cui iniziano i trasferimenti pianificati.

## Informazioni su questa attività

Se si desidera utilizzare IBM MQ Explorer per monitorare i trasferimenti associati a un gestore code di coordinamento su un sistema remoto, seguire le istruzioni nell'argomento [“Configurazione di IBM MQ Explorer per monitorare un gestore code di coordinamento remoto”](#) a pagina 214.

Le informazioni sul trasferimento file precedenti non vengono conservate dopo l'arresto e il riavvio di IBM MQ Explorer. Al riavvio, le informazioni sui trasferimenti passati vengono cancellate dalla scheda

**Avanzamento trasferimento corrente** . È possibile cancellare i trasferimenti completati utilizzando l'icona **Rimuovi trasferimenti completati**  in qualsiasi momento quando IBM MQ Explorer è aperto.

## Procedura

Dopo aver avviato un nuovo trasferimento file utilizzando IBM MQ Explorer o la riga comandi, è possibile monitorare l'avanzamento del trasferimento nella scheda **Avanzamento trasferimento corrente** . Per ogni trasferimento in corso vengono visualizzate le seguenti informazioni:

- a) **Origine**. Il nome dell'agente utilizzato per trasferire il file dal sistema di origine.
- b) **Destinazione**:. Il nome dell'agente utilizzato per ricevere il file sul sistema di destinazione.
- c) **File corrente**. Il nome del file attualmente trasferito. La parte del singolo file che è già stato trasferito viene visualizzata in B, KiB, MiB, GiBo TiB insieme alla dimensione totale del file tra parentesi. L'unità di misura visualizzata dipende dalla dimensione del file.  
B è byte al secondo. KiB/s è kibibyte al secondo, dove 1 kibibyte è uguale a 1024 byte. MiB/s è mebibyte al secondo, dove 1 mebibyte equivale a 1 048 576 byte. GiB/s è gibibyte al secondo dove 1 gibibyte è uguale a 1 073 741 824 byte. TiB/s è tebibyte al secondo dove 1 tebibyte equivale a 1 099 511 627 776 byte.
- d) **Numero file**. Se si sta trasferendo più di un file, questo numero rappresenta la distanza del gruppo totale di file da trasferire.
- e) **Avanzamento**. La barra di avanzamento mostra il completamento del trasferimento file corrente come percentuale.
- f) **Tasso**. La frequenza con cui il file viene trasferito in KiB/s (kibibyte al secondo, dove 1 kibibyte equivale a 1024 byte).
- g) **Avviato (fuso orario selezionato)**. L'ora in cui è stato avviato il trasferimento file, presentata nel fuso orario selezionato dell'amministratore. Per modificare il fuso orario visualizzato, selezionare **Finestra > Preferenze > IBM MQ Explorer > Trasferimento file gestito** e selezionare un fuso orario alternativo dall'elenco **Fuso orario**: . Fare clic su **OK**.

Se il trasferimento entra in uno stato di ripristino durante il trasferimento del file, l'ora di inizio viene aggiornata in modo da riflettere l'ora in cui è stato ripreso il trasferimento del file.

## Risultati

Questa scheda aggiorna regolarmente le informazioni automaticamente, ma per forzare una vista aggiornata di ciò che viene visualizzato nella scheda **Avanzamento trasferimento corrente** , fare clic su **Aggiorna**  nella barra degli strumenti della vista Contenuto.

Per eliminare trasferimenti file dalla scheda **Avanzamento trasferimento corrente** , fare clic su **Rimuovi trasferimenti completati**  nella barra degli strumenti della vista Contenuto. Facendo clic su questo pulsante si rimuovono i dettagli del trasferimento file solo dalla scheda; non si arresta o si annulla un trasferimento corrente o pianificato.

Se si desidera tornare alla scheda **Avanzamento trasferimento corrente** dopo averlo chiuso, è possibile visualizzare la scheda facendo clic su **Finestra > Mostra vista > Altro > Altro > Trasferimento file gestito - Avanzamento trasferimento corrente**. Fare clic su **OK**.

## Operazioni successive

Inoltre, è possibile sviluppare applicazioni per il monitoraggio del trasferimento file personalizzato. Ciò può essere realizzato creando una sottoscrizione all'argomento di gestione Managed File Transfer appropriato (in modo programmatico o amministrativo) e l'applicazione di monitoraggio può quindi ricevere le pubblicazioni dell'attività di trasferimento file Managed File Transfer sull'argomento. Per ulteriori informazioni sull'argomento della sottoscrizione e sul formato del messaggio di pubblicazione, consultare [Esempi di messaggi di avanzamento del trasferimento file](#).

## Attività correlate

[“Configurazione di IBM MQ Explorer per monitorare un gestore code di coordinamento remoto” a pagina 214](#)

Utilizzare IBM MQ Explorer per monitorare i trasferimenti di file associati a un gestore code di coordinamento in esecuzione su un sistema remoto. In IBM WebSphere MQ 7.5, o versioni successive, è necessario un sistema in grado di eseguire IBM MQ Explorer. Il componente IBM MQ Explorer deve essere installato per essere in grado di connettersi al gestore code di coordinamento remoto.

[“Visualizzazione dello stato dei trasferimenti file nel log di trasferimento” a pagina 215](#)

È possibile visualizzare i dettagli dei trasferimenti file utilizzando il **Log trasferimenti** in IBM MQ Explorer. Questi possono essere trasferimenti avviati dalla riga comandi o da IBM MQ Explorer. È anche possibile personalizzare quanto visualizzato nel **Log di trasferimento**.

## Configurazione di IBM MQ Explorer per monitorare un gestore code di coordinamento remoto

Utilizzare IBM MQ Explorer per monitorare i trasferimenti di file associati a un gestore code di coordinamento in esecuzione su un sistema remoto. In IBM WebSphere MQ 7.5, o versioni successive, è necessario un sistema in grado di eseguire IBM MQ Explorer. Il componente IBM MQ Explorer deve essere installato per essere in grado di connettersi al gestore code di coordinamento remoto.

## Informazioni su questa attività

Presupposti: autorizzazione a connettersi al gestore code di coordinamento remoto configurando il gestore code per consentire le connessioni remote.

Per ulteriori informazioni su come configurarlo, consultare [Connessione a un gestore code in modalità client con autenticazione di canale](#) e [Gestione delle autorizzazioni per le risorse specifiche di MFT](#).

Per monitorare i gestori code e i trasferimenti file tra agent su un sistema su cui non è in esecuzione Windows o Linux, configurare IBM MQ Explorer per la connessione al sistema remoto utilizzando la seguente procedura:

## Procedura

1. Avviare il IBM MQ Explorer locale.
2. Quando IBM MQ Explorer viene caricato, fare clic con il pulsante destro del mouse sulla cartella **Managed File Transfer** e selezionare **Nuova configurazione**.
3. Procedere con la procedura guidata, selezionando il gestore code Coordinamento e Comandi, quindi definire un nome per la configurazione.
4. Fare clic su **Fine** per completare la definizione.
5. Una volta terminata la definizione, fare clic con il tasto destro del mouse sulla definizione e selezionare **Connetti**.

## Risultati

Ora avviare IBM MQ Explorer e utilizzarlo per monitorare l'attività di trasferimento per la rete Managed File Transfer associata con il gestore code di coordinamento.

## Attività correlate

[“Monitoraggio dei trasferimenti file in corso” a pagina 212](#)

È possibile monitorare un trasferimento file in corso utilizzando la scheda **Managed File Transfer - Avanzamento trasferimento corrente** in IBM MQ Explorer. Questo trasferimento file può essere avviato da IBM MQ Explorer o dalla riga comandi. La scheda visualizza anche l'avanzamento dei trasferimenti pianificati nel momento in cui iniziano i trasferimenti pianificati.

[“Visualizzazione dello stato dei trasferimenti file nel log di trasferimento” a pagina 215](#)

È possibile visualizzare i dettagli dei trasferimenti file utilizzando il **Log trasferimenti** in IBM MQ Explorer. Questi possono essere trasferimenti avviati dalla riga comandi o da IBM MQ Explorer. È anche possibile personalizzare quanto visualizzato nel **Log di trasferimento**.

## Visualizzazione dello stato dei trasferimenti file nel log di trasferimento

È possibile visualizzare i dettagli dei trasferimenti file utilizzando il **Log trasferimenti** in IBM MQ Explorer. Questi possono essere trasferimenti avviati dalla riga comandi o da IBM MQ Explorer. È anche possibile personalizzare quanto visualizzato nel **Log di trasferimento**.

### Procedura

1. Espandere **Managed File Transfer** nella vista Navigator ed espandere il nome del gestore code di coordinamento per cui si desidera visualizzare il log di trasferimento.
2. Fare clic su **Log trasferimento** nella vista Navigator . Il **Log di trasferimento** viene visualizzato nella vista Contenuto.
3. La finestra **Log trasferimenti** visualizza i seguenti dettagli sui trasferimenti file:
  - a) **Origine** Il nome dell'agent sul sistema in cui si trova il file di origine.
  - b) **Destinazione** il nome dell'agent sul sistema a cui si desidera trasferire il file.
  - c) **Stato di completamento** Lo stato del trasferimento file. Lo stato può avere uno dei seguenti valori: "Avviato", "In corso", "Riuscito", "Parzialmente riuscito", "Annullato" o "Non riuscito".
  - d) **Proprietario** L'ID utente sull'host che ha inoltrato la richiesta di trasferimento.
  - e) **Avviato (fuso orario selezionato)** La data/ora in cui la richiesta di trasferimento file è stata accettata dall'agent Managed File Transfer , presentata nel fuso orario selezionato dell'amministratore. Per modificare il fuso orario visualizzato, fare clic su **Finestra > Preferenze > IBM MQ Explorer > Managed File Transfer** e selezionare un fuso orario alternativo dall'elenco **Fuso orario:** . Fare clic su **OK**.
  - f) **Stato registrato (fuso orario selezionato)** (Questa colonna non è visualizzata per default. È possibile scegliere di visualizzare la colonna utilizzando la finestra **Configura colonne log di trasferimento**  . La data e l'ora in cui è stato registrato lo stato di completamento, nel fuso orario selezionato dall'amministratore.
  - g) **Nome lavoro** Un identificativo specificato da un utente utilizzando il parametro **-jn di fteCreateTransfer** o in uno script Ant
  - h) **ID trasferimento** L'identificativo univoco per il trasferimento file.
  - i) **Connessione: diretta** Sono elencati i dettagli su **Numero processo, Nome processo, Nodo primario, Nodo secondario, Tipo di origine e Tipo di destinazione** .

### Risultati

**Nota:** Il formato interno del log di trasferimento è stato modificato in IBM MQ 8.0.0 Fix Pack 1 per APAR IC99545. Di conseguenza, se un IBM MQ Explorer viene aggiornato a V8.0.0.1 o versioni successive e quindi ripristinato a V8.0.0.0, non viene visualizzato alcun XML di verifica per i trasferimenti che hanno avuto luogo mentre IBM MQ Explorer si trovava in V8.0.0.1. Il pannello XML nella finestra **Proprietà** per questi trasferimenti conterrà una finestra di testo vuota.

Per visualizzare ulteriori dettagli su un trasferimento completato, espandere il trasferimento a cui si è interessati facendo clic sul segno più (+). È quindi possibile visualizzare tutti i nomi file di origine e di destinazione inclusi in tale trasferimento. Tuttavia, se il trasferimento è attualmente in corso ed è composto da molti file, è possibile visualizzare solo i file che sono già stati trasferiti fino ad ora.

Per aggiornare quanto visualizzato nel **Log di trasferimento**, fare clic sul pulsante **Aggiorna**  nella barra degli strumenti della vista Contenuto. Le informazioni sul trasferimento file nel log di trasferimento rimangono nel log dopo l'arresto e il riavvio di IBM MQ Explorer. Se si desidera eliminare tutti i

trasferimenti file completati dal log, fare clic su **Rimuovi trasferimenti completati**  nella barra degli strumenti della vista Contenuto.

Per eliminare un singolo trasferimento file completato dal log, fare clic con il tasto destro del mouse sul trasferimento e fare clic su **Elimina**. Se si elimina un trasferimento, non si arresta o si annulla un trasferimento in corso o che è stato pianificato; si stanno eliminando solo i dati cronologici memorizzati.

Per copiare l'identificativo univoco di un trasferimento negli appunti, fare clic con il pulsante destro del mouse su tale trasferimento e fare clic su **Copia ID**.

I metadati e l'XML di controllo completo per il trasferimento sono disponibili dal menu a comparsa, nell'azione **Proprietà**.

### Attività correlate

[“Monitoraggio dei trasferimenti file in corso” a pagina 212](#)

È possibile monitorare un trasferimento file in corso utilizzando la scheda **Managed File Transfer - Avanzamento trasferimento corrente** in IBM MQ Explorer. Questo trasferimento file può essere avviato da IBM MQ Explorer o dalla riga comandi. La scheda visualizza anche l'avanzamento dei trasferimenti pianificati nel momento in cui iniziano i trasferimenti pianificati.

[“Configurazione del log di trasferimento” a pagina 216](#)

È possibile configurare quali informazioni vengono visualizzate e come vengono visualizzate nel **Log di trasferimento** in IBM MQ Explorer.

[“Impostazione di una scadenza per il recupero di trasferimenti in stallo” a pagina 301](#)

È possibile impostare un timeout di ripristino del trasferimento per i trasferimenti di file in stallo che si applica a tutti i trasferimenti per un agent di origine. È anche possibile impostare un timeout di ripristino trasferimento per un singolo trasferimento. Se si imposta un periodo di tempo specifico, in secondi, durante il quale un agent di origine tenta di ripristinare un trasferimento file in stallo e il trasferimento non riesce quando l'agent raggiunge il timeout, il trasferimento non riesce.

## Configurazione del log di trasferimento

È possibile configurare quali informazioni vengono visualizzate e come vengono visualizzate nel **Log di trasferimento** in IBM MQ Explorer.

### Informazioni su questa attività

Per ridisporre l'ordine delle colonne nel **Log di trasferimento**, fare clic sul titolo della colonna che si desidera spostare e trascinare la colonna nella sua nuova posizione. Il nuovo ordine delle colonne viene conservato solo fino al successivo arresto e riavvio di IBM MQ Explorer.

Per filtrare le voci nel **Log di trasferimento**, immettere una stringa nel campo **Filtra le voci di log visualizzate**. Per ripristinare tutte le voci nel log, eliminare la stringa immessa dal campo. È possibile utilizzare qualsiasi espressione regolare Java valida in questo campo. Per ulteriori informazioni, consultare [Espressioni regolari utilizzate da MFT](#).

Per personalizzare le colonne visualizzate nel log di trasferimento, utilizzare **Configura colonne log di trasferimento** . Utilizzare la seguente procedura per avviare ed utilizzare la finestra **Configura colonne log di trasferimento**.

### Procedura

1. Assicurarsi di avere il **Log di trasferimento** aperto nella vista Contenuto. Fare clic su **Configura colonne log di trasferimento**  sulla barra degli strumenti della vista Contenuto. Viene visualizzata la finestra **Configura colonne log di trasferimento**.
2. Per personalizzare la vista del **Log di trasferimento**, selezionare o deselezionare le singole caselle di spunta per le colonne che si desidera visualizzare o nascondere. È possibile fare clic su **Seleziona tutto**, quindi su **OK** per selezionare tutte le caselle di spunta oppure su **Deseleziona tutto**, quindi su **OK** per deselezionare tutte le caselle di spunta.

## Attività correlate

“Monitoraggio dei trasferimenti file in corso” a pagina 212

È possibile monitorare un trasferimento file in corso utilizzando la scheda **Managed File Transfer - Avanzamento trasferimento corrente** in IBM MQ Explorer. Questo trasferimento file può essere avviato da IBM MQ Explorer o dalla riga comandi. La scheda visualizza anche l'avanzamento dei trasferimenti pianificati nel momento in cui iniziano i trasferimenti pianificati.

“Visualizzazione dello stato dei trasferimenti file nel log di trasferimento” a pagina 215

È possibile visualizzare i dettagli dei trasferimenti file utilizzando il **Log trasferimenti** in IBM MQ Explorer. Questi possono essere trasferimenti avviati dalla riga comandi o da IBM MQ Explorer. È anche possibile personalizzare quanto visualizzato nel **Log di trasferimento**.

## Monitoraggio delle risorse MFT

È possibile monitorare le risorse Managed File Transfer ; ad esempio, una coda o una directory. Quando viene soddisfatta una condizione su questa risorsa, il monitoraggio risorse avvia un'attività, come un trasferimento file. È possibile creare un monitoraggio delle risorse utilizzando il comando **ftCreateMonitor** o la vista **Monitor** nel plug-in Managed File Transfer per IBM MQ Explorer.

### Informazioni su questa attività

Il monitoraggio delle risorse Managed File Transfer usa la terminologia seguente:

#### Controllo risorse

Un monitoraggio risorse è un processo che esegue il polling di una risorsa (come un indirizzario o una coda) a un intervallo regolare predefinito per verificare se il contenuto della risorsa è stato modificato. In tal caso, il contenuto viene confrontato con la serie di condizioni per questo monitor. Se esiste una corrispondenza, l'attività per questo monitoraggio viene avviata.

#### risorsa

La risorsa di sistema che il monitoraggio risorse esamina ad ogni intervallo di polling da confrontare con le condizioni di trigger. Le code, le directory o le strutture di directory nidificate possono essere la risorsa monitorata.

#### Condizione e condizione trigger

Una condizione è un'espressione che viene valutata (generalmente rispetto al contenuto della risorsa monitorata). Se l'espressione viene valutata true, la condizione contribuisce alla condizione di trigger generale.

La condizione trigger è la condizione generale, che viene soddisfatta quando tutte le condizioni vengono soddisfatte. Quando la condizione di trigger viene soddisfatta, l'attività può procedere.

#### Attività

Un'attività è l'operazione che viene avviata quando viene soddisfatta la condizione di trigger o la serie di condizioni. Le attività supportate sono il trasferimento file e la chiamata del comando.

#### File trigger

Un file trigger è un file posizionato in una directory monitorata per indicare che un'attività (di solito un trasferimento) può iniziare. Ad esempio, potrebbe indicare che tutti i file da elaborare sono arrivati in un'ubicazione nota e possono essere trasferiti o altrimenti gestiti. Il nome del file trigger può essere utilizzato per specificare i file da trasferire utilizzando la sostituzione variabile. Per ulteriori informazioni, consultare [“Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile”](#) a pagina 228.

Il file trigger è noto anche come file pronto o file go. Tuttavia, in questa documentazione viene generalmente indicato come file trigger.

Il monitoraggio delle risorse non è supportato sugli agent bridge di protocollo o sugli Connect:Direct agent bridge.

## Concetti di monitoraggio delle risorse MFT

Una panoramica dei concetti chiave della funzione di monitoraggio delle risorse Managed File Transfer .

## Monitoraggi risorse

Si crea un controllo risorse utilizzando il comando **fteCreateMonitor**, che crea e avvia un nuovo controllo risorse dalla riga comandi. Il monitoraggio delle risorse è associato a un agent Managed File Transfer ed è attivo solo quando tale agent è in esecuzione. Quando l'agent di monitoraggio viene arrestato, lo stesso vale per il monitoraggio delle risorse. Se l'agent è già in esecuzione quando viene creato il monitoraggio risorse, il monitoraggio risorse viene avviato immediatamente. L'agente di controllo deve essere anche l'agente di origine dell'attività avviata dal controllo risorse.

I nomi del monitoraggio delle risorse devono essere univoci all'interno del relativo agent. Il nome del controllo risorse deve avere una lunghezza minima di un carattere e non deve contenere caratteri asterisco (\*), percentuale (%) o punto interrogativo (?). Il caso in cui viene fornito un nome di monitoraggio risorse viene ignorato e il nome di monitoraggio risorse viene convertito in maiuscolo. Se si tenta di creare un monitoraggio risorse con un nome già presente, la richiesta viene ignorata e il tentativo viene registrato nell'argomento del log del monitoraggio risorse.

**Nota:** Non è possibile creare un monitoraggio risorse con una definizione attività che contenga trasferimenti pianificati.

**V 9.2.2** Per Long Term Supporte per Continuous Delivery prima di IBM MQ 9.2.2, l'unico modo per arrestare un monitoraggio risorse consiste nell'arrestare l'agent che sta eseguendo l'operazione di monitoraggio. Per riavviare un monitoraggio risorse, è necessario riavviare l'agente. Da IBM MQ 9.2.2, è possibile avviare e arrestare i monitoraggi delle risorse senza dover arrestare o riavviare un agent. Per ulteriori informazioni, consultare [“Avvio e arresto dei monitoraggi delle risorse”](#) a pagina 221.

Non esiste alcuna limitazione sul numero di monitoraggi delle risorse che possono essere creati su un agent e tutti vengono eseguiti con la stessa priorità. Considerare le implicazioni della sovrapposizione delle risorse monitorate, le condizioni di trigger in conflitto e la frequenza con cui viene eseguito il polling delle risorse.

La sovrapposizione dei monitoraggi delle risorse può causare:

- Possibile conflitto sull'ubicazione/elementi di origine.
- Possibili richieste di trasferimento duplicate per gli stessi elementi di origine.
- Errori o errori imprevisti per i trasferimenti dovuti a conflitti di elementi di origine.

Se più monitoraggi eseguono la scansione della stessa ubicazione e possono essere attivati sugli stessi elementi, è possibile che si verifichi il problema di due monitoraggi differenti che inoltrano le richieste di trasferimento gestito per lo stesso elemento.

I monitoraggi risorse esaminano il contenuto delle risorse dopo ogni periodo di intervallo di polling. Il contenuto della risorsa viene confrontato con le condizioni di trigger e se tali condizioni vengono soddisfatte, viene richiamata l'attività associata al monitoraggio risorse.

L'attività viene avviata in modo asincrono. Se esiste una corrispondenza di condizioni e l'attività viene avviata, il monitoraggio delle risorse continua a eseguire il polling per ulteriori modifiche al contenuto della risorsa. Quindi, ad esempio, se si verifica una corrispondenza perché un file denominato `reports.go` è arrivato in una directory monitorata, l'attività verrà avviata una sola volta. Al successivo intervallo di polling, anche se il file esiste ancora, l'attività non viene riavviata. Tuttavia, se il file viene eliminato e quindi riposizionato nella directory o se il file viene aggiornato (in modo che l'attributo della data dell'ultima modifica venga modificato), il successivo controllo della condizione di trigger fa sì che l'attività venga richiamata di nuovo.

Prima di IBM MQ 9.1.5, se un monitoraggio delle risorse esegue un polling che impiega più tempo dell'intervallo di polling, ciò significa che il successivo polling inizia non appena quello corrente termina senza interlinea, il che potrebbe avere un effetto sulla velocità con cui i monitoraggi delle risorse inoltrano il lavoro a un agent. Ciò potrebbe causare problemi di prestazioni se gli elementi rilevati durante il primo polling sono ancora presenti quando si verifica il secondo.

**V 9.2.0** Da IBM MQ 9.1.5, il monitoraggio risorse utilizza il servizio ScheduledExecutore avvia il successivo polling solo dopo il completamento del polling precedente più il periodo di intervallo di polling configurato. Ciò significa che ci sarà sempre un divario tra gli intervalli di polling, piuttosto che avere un

altro polling che inizia subito dopo il polling precedente se il tempo di polling era più lungo dell'intervallo di polling.

**V 9.2.0** Da IBM MQ 9.1.3, se il trasferimento di un file non è riuscito, è possibile cancellare la cronologia del controllo risorse, che consente l'inoltro di un'altra richiesta di trasferimento senza la necessità di eliminare il file e inserirlo nuovamente nell'indirizzario oppure aggiornare il file per modificare l'attributo della data dell'ultima modifica. La cancellazione della cronologia è utile, ad esempio, in situazioni in cui è necessario trasferire il file, ma la modifica del file non è possibile. Per ulteriori informazioni, consultare [“Cancellazione della cronologia del controllo risorse”](#) a pagina 245.

## Risorse

I monitoraggi delle risorse in Managed File Transfer possono eseguire il polling del contenuto dei due seguenti tipi di risorse:

### Directory o strutture di directory nidificate

Uno scenario comune è quello di monitorare una directory per la presenza di un file trigger. Un'applicazione esterna potrebbe elaborare più file e collocarli in una directory di origine nota. Quando l'applicazione ha completato la sua elaborazione, indica che i file sono pronti per essere trasferiti o su cui si agisce in altro modo, posizionando un file trigger in un'ubicazione monitorata. Il file trigger può essere rilevato da un controllo risorse Managed File Transfer e il trasferimento di tali file dalla directory di origine a un altro Managed File Transfer Agent viene avviato.

Per impostazione predefinita, la directory specificata viene monitorata. Per esaminare anche le sottodirectory, impostare il livello di ricorsione nel comando **fteCreateTransfer**.

Di seguito sono riportati due esempi di monitoraggio di una directory:

- Monitorare un file trigger (ad esempio, `trigger.file`) e quindi trasferire un carattere jolly (ad esempio, `*.zip`).
- Monitorare `*.zip` e quindi trasferire `${FilePath}` (ad esempio, il file che ha attivato il trasferimento). Per ulteriori informazioni sulla sostituzione delle variabili, consultare [“Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile”](#) a pagina 228.

**Nota:** Non creare un monitor che monitori `*.zip` e quindi trasferisca `*.zip`. Il monitoraggio tenta di avviare un trasferimento di `*.zip` per ogni file `.zip` sul sistema. Vale a dire, il controllo genera \* numero di trasferimenti per `*.zip`.

Per un esempio di creazione di un monitoraggio risorse per monitorare una directory, consultare [“Monitoraggio di una directory e utilizzo della sostituzione della variabile”](#) a pagina 226.

### Code IBM MQ

Un esempio di monitoraggio di una coda è che un'applicazione esterna potrebbe generare messaggi e posizionarli su una coda nota con lo stesso ID gruppo. Quando l'applicazione ha completato l'inserimento dei messaggi nella coda, indica che il gruppo è completo. Il gruppo completo di messaggi può essere rilevato da un monitoraggio risorse Managed File Transfer e viene avviato il trasferimento del gruppo di messaggi dalla coda di origine a un file. Per un esempio di creazione di un controllo risorse per monitorare una coda, consultare [“Esempio: configurazione di una risorsa MFT”](#) a pagina 228.

**Nota:** È possibile specificare un solo monitor per coda. Se si specifica più di un monitor per eseguire il polling di una coda IBM MQ, si verifica un comportamento imprevedibile.

Il monitoraggio dei dataset non è supportato.

## Condizioni e condizioni di attivazione

La condizione viene soddisfatta quando la risorsa contiene un valore che corrisponde a qualche altra stringa o modello. Le condizioni possono essere una delle seguenti:

- Corrispondenza sul nome file (modello)
- Nessuna corrispondenza sul nome file (modello)

- Dimensione file
- Corrisponde se la dimensione del file rimane la stessa per un certo numero di polling

La corrispondenza del nome file può essere espressa come:

- Corrispondenza stringa esatta
- Corrispondenza di caratteri jolly semplici come descritto in [Utilizzo di caratteri jolly con MFT](#)
- corrispondenza espressione regolare

I nomi file possono anche essere esclusi dalla corrispondenza dei nomi file utilizzando un carattere jolly o un'espressione regolare Java che identifica i nomi file che non corrispondono mai.

Quando viene rilevato un file corrispondente, viene conservata la data / ora dell'ultima modifica. Se sondaggi successivi rilevano che il file è stato modificato, la condizione di trigger viene soddisfatta nuovamente e l'attività viene avviata. Se la condizione è rilevare quando un file non esiste, se nessun file nella directory monitorata corrisponde al modello del nome file, l'attività viene avviata. Se un file viene aggiunto alla directory che non corrisponde al pattern del nome file, l'attività viene avviata solo se il file viene eliminato.

## Attività

Managed File Transfer supporta i seguenti due tipi di attività che è possibile configurare per essere avviati dai monitoraggi delle risorse:

### Attività di trasferimento file

Le attività di trasferimento file sono definite allo stesso modo di qualsiasi altro trasferimento file. Un modo utile per generare l'attività XML richiesta da un controllo consiste nell'eseguire il comando `fteCreateTransfer` con il parametro **-gt**. Questo comando genera una definizione di attività come documento XML, inclusa la specifica di trasferimento. Si passa quindi il nome del documento XML dell'attività come valore per il parametro **-mt** nel comando `fteCreateMonitor`. Quando **fteCreateMonitor** viene eseguito, legge il documento XML dell'attività. Dopo l'esecuzione di **fteCreateMonitor**, le modifiche apportate al file XML dell'attività non vengono utilizzate dal monitor.

Quando si utilizza un'attività di trasferimento file, è possibile selezionare quante condizioni di trigger vengono raggruppate in un'attività. Il valore predefinito è per una condizione trigger per avviare un'attività. È possibile eseguire il comando `fteCreateMonitor` con l'opzione **-bs** per selezionare il numero di condizioni di trigger raggruppate in un'unica attività.

### Attività di comando

Le attività di comando possono eseguire script Ant, richiamare programmi eseguibili o eseguire lavori JCL. Per ulteriori informazioni, consultare [“Configurazione delle attività di monitoraggio MFT per avviare comandi e script”](#) a pagina 222.

## File trigger

È possibile utilizzare il contenuto di un file trigger in un controllo risorse per definire una serie di file da trasferire in una richiesta di trasferimento singola. Ogni volta che viene rilevato un file trigger corrispondente, il relativo contenuto viene analizzato per i percorsi file di origine e facoltativamente per i percorsi file di destinazione. Questi percorsi file vengono quindi utilizzati per definire elementi file nel file XML di trasferimento attività specificato, inoltrato come singola richiesta di trasferimento all'agent. La definizione del monitoraggio risorse determina se il contenuto del trigger è abilitato.

Il formato di ogni file trigger è un singolo percorso file da trasferire su ogni riga di testo. Il formato predefinito per la riga è un singolo percorso file di origine o un percorso file di origine e di destinazione separati da una virgola.

Per ulteriori informazioni ed esempi, consultare [“Utilizzo di un file trigger”](#) a pagina 237.

## Avvio e arresto dei monitoraggi delle risorse

Per Long Term Supporte per Continuous Delivery prima di IBM MQ 9.2.2, l'unico modo per arrestare un monitoraggio risorse consiste nell'arrestare l'agent che sta eseguendo l'operazione di monitoraggio. Per riavviare un monitoraggio risorse, è necessario riavviare l'agente. Per ulteriori informazioni, consultare [“Avvio di un agent MFT” a pagina 200](#) e [“Arresto di un agent MFT” a pagina 205](#).

**V 9.2.2** Da IBM MQ 9.2.2, è possibile avviare e arrestare i monitoraggi risorse senza dover arrestare o riavviare un agent utilizzando i comandi **fteStartMonitor** e **fteStopMonitor**. Ciò è utile, ad esempio, nelle situazioni seguenti:

- Se un agent dispone di più monitoraggi delle risorse e solo alcuni di essi hanno rilevato errori, ma i restanti monitoraggi delle risorse stanno ancora funzionando correttamente, si desidera solo riavviare i monitoraggi delle risorse non riusciti.
- Se si desidera arrestare un controllo risorse per eseguire del lavoro di manutenzione o se il controllo risorse non è richiesto per un determinato periodo di tempo e non si desidera che venga eseguito inutilmente, consumando risorse di sistema preziose.

**V 9.2.2** Per ulteriori informazioni, consultare [Avvio di un monitoraggio risorse MFT](#) e [Arresto di un monitoraggio risorse MFT](#).

**V 9.2.2**

Comando	Comportamento del monitoraggio risorse
<b>V 9.2.2</b> <b>fteStartMonitor</b>	Se l'agent è in esecuzione, il monitoraggio risorse viene avviato se è attualmente arrestato.
<b>V 9.2.2</b> <b>fteStopMonitor</b>	Se l'agent è in esecuzione, il monitoraggio delle risorse viene arrestato se è attualmente avviato.
<b>fteStartAgent</b>	Il monitoraggio risorse viene avviato come parte dell'avvio dell'agente, indipendentemente dalle precedenti chiamate a <b>fteStopMonitor</b> .
<b>fteStopAgent</b>	Tutti i monitoraggi delle risorse in esecuzione vengono arrestati.

## Backup e ripristino dei monitoraggi risorse

È possibile eseguire il backup dei monitoraggi delle risorse già definiti in modo da poterli riutilizzare in futuro. Esistono varie opzioni che è possibile utilizzare come segue:

- Utilizzare il comando **fteCreateMonitor** con il parametro **-ox** per esportare una configurazione del monitoraggio risorse in un file XML e con il parametro **-ix** per ripristinare un monitoraggio risorse importando la configurazione del monitoraggio risorse da un file XML.
- Utilizzare il comando **fteListMonitors** con **-ox** per esportare la definizione per un singolo monitoraggio risorse in un file XML.
- Utilizzare il comando **fteListMonitors** con il **-od** per esportare più definizioni di monitoraggio risorse in una directory specificata. Ogni definizione di monitoraggio risorse viene salvata in un file XML separato. È anche possibile utilizzare l'opzione **-od** per esportare una singola definizione di monitoraggio risorse in una directory specificata.

Per ulteriori informazioni, consultare [“Backup e ripristino dei monitoraggi delle risorse MFT” a pagina 244](#).

## Registrazione monitoraggio risorse

Da IBM MQ 9.1.0, Managed File Transfer include la registrazione del monitoraggio risorse. Per ulteriori informazioni, consultare [“Registrazione dei monitoraggi delle risorse MFT” a pagina 239](#).

## Concetti correlati

[“Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile” a pagina 228](#)

Quando le condizioni del trigger di un controllo risorse attivo vengono soddisfatte, viene richiamata l'attività definita. Oltre a richiamare l'attività di trasferimento o comando con lo stesso agent di destinazione o lo stesso nome file di destinazione ogni volta, è anche possibile modificare la definizione dell'attività in fase di runtime. A tale scopo, inserire i nomi delle variabili nel file XML di definizione attività. Quando il controllo determina che le condizioni di trigger sono soddisfatte e che la definizione dell'attività contiene nomi di variabili, sostituisce i nomi di variabili con valori di variabile e richiama l'attività.

## Attività correlate

[“Configurazione delle attività di monitoraggio MFT per avviare comandi e script” a pagina 222](#)

I monitoraggi risorse non sono limitati all'esecuzione di trasferimenti file come attività associata. È anche possibile configurare il monitor per richiamare altri comandi dall'agent di monitoraggio, inclusi i programmi eseguibili, gli script Ant o i lavori JCL. Per richiamare i comandi, modificare l'XML di definizione dell'attività di monitoraggio in modo da includere uno o più elementi di comando con i parametri di chiamata del comando corrispondenti, ad esempio argomenti e proprietà.

[“Esempio: configurazione di una risorsa MFT” a pagina 228](#)

È possibile specificare una coda IBM MQ come la risorsa che deve essere monitorata da un monitoraggio risorse utilizzando il parametro **-mq** con il comando **fteCreateMonitor**.

[“Monitoraggio di una coda e utilizzo della sostituzione di variabili” a pagina 234](#)

È possibile monitorare una coda e trasferire messaggi dalla coda monitorata a un file utilizzando il comando **fteCreateMonitor**. Il valore di qualsiasi proprietà del messaggio IBM MQ nel primo messaggio da leggere dalla coda monitorata può essere sostituito nella definizione XML dell'attività e utilizzato per definire il funzionamento del trasferimento.

## Riferimenti correlati

**fteCreateMonitor**: crea un monitoraggio risorse MFT

[fteListMonitoraggi](#): elenco MFT monitoraggi risorse

[Controllo fteDelete](#): elimina un controllo risorse MFT

## Configurazione delle attività di monitoraggio MFT per avviare comandi e script

I monitoraggi risorse non sono limitati all'esecuzione di trasferimenti file come attività associata. È anche possibile configurare il monitor per richiamare altri comandi dall'agent di monitoraggio, inclusi i programmi eseguibili, gli script Ant o i lavori JCL. Per richiamare i comandi, modificare l'XML di definizione dell'attività di monitoraggio in modo da includere uno o più elementi di comando con i parametri di chiamata del comando corrispondenti, ad esempio argomenti e proprietà.

## Informazioni su questa attività

Il percorso file del programma eseguibile, dello script Ant o del lavoro JCL che si desidera venga richiamato dall'agent di monitoraggio deve essere incluso nel `commandPath` dell'agent di monitoraggio. Per informazioni sulla proprietà del percorso del comando, consultare [commandPath MFT property](#).

È possibile creare il documento XML di definizione attività in uno dei seguenti modi:

- Creare manualmente il documento XML di definizione attività in base allo schema `FileTransfer.xsd`.
- Utilizzare un documento XML generato come base per la propria definizione di attività.

Se si desidera un'attività di trasferimento o un'attività di comando, la definizione dell'attività deve iniziare con un elemento root `<request>`. L'elemento child di `<request>` deve essere `<managedTransfer>` o `<managedCall>`. Generalmente, è possibile scegliere `<managedCall>` quando è presente un singolo comando o script da eseguire e `<managedTransfer>` se si desidera che l'attività includa un trasferimento file e, facoltativamente, fino a quattro chiamate di comando.

## Procedura

- Per creare manualmente il documento XML di definizione attività in base allo schema `FileTransfer.xsd`, consultare [“Creazione manuale di un XML di definizione attività in base allo schema” a pagina 223](#).
- Per creare una definizione di attività modificando un documento generato, modificare il documento XML generato dal parametro `fteCreateTransfer -gt`. Per ulteriori informazioni, consultare [“Creazione di un documento di definizione attività modificando un documento generato” a pagina 225](#).

### **Creazione manuale di un XML di definizione attività in base allo schema**

È possibile creare manualmente un file XML di definizione attività in base allo schema `FileTransfer.xsd`.

### **Informazioni su questa attività**

Lo schema `FileTransfer.xsd` è disponibile in `MQ_INSTALLATION_PATH/mqft/samples/schema`. Per ulteriori informazioni su questo schema, consultare [Formato del messaggio di richiesta di trasferimento file](#).

### **Esempio**

Il seguente esempio mostra un documento XML di definizione attività di esempio salvato come `cleanuptask.xml`, che utilizza l'elemento `<managedCall>` per richiamare uno script Ant denominato `RunCleanup.xml`. Lo script `RunCleanup.xml` Ant deve trovarsi sul `commandPath` dell'agent di monitoraggio.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <target>copy_to_archive</target>
          <target>rename_temps</target>
          <target>delete_files</target>
          <property name="trigger.filename" value="{FileName}"/>
          <property name="trigger.path" value="{FilePath}"/>
        </command>
      </call>
    </transferSet>
  </managedCall>
</request>
```

L'elemento `<agent>` specifica il Managed File Transfer Agent configurato con lo script Ant denominato sul relativo `commandPath`.

La struttura `<call><command> . . .` definisce l'eseguibile o lo script che si desidera eseguire. Il comando utilizza un attributo `type` facoltativo che può avere uno dei seguenti valori:

#### **antscript**

Eseguire uno script di Ant in una JVM separata.

## Eseguibile

Richiamare un programma eseguibile.

## JCL

Richiamare un lavoro JCL.

Se si omette l'attributo `type`, viene utilizzato il valore predefinito eseguibile.

L'attributo `name` specifica il nome dello script, dell'eseguibile o del lavoro JCL Ant che si desidera eseguire, senza alcuna informazione sul percorso. L'agent ricerca lo script o il programma nelle ubicazioni specificate dalla proprietà `commandPath` nel file `agent.properties` dell'agent.

L'attributo `retrycount` specifica il numero di volte in cui provare a richiamare di nuovo il programma se il programma non restituisce un codice di ritorno di esito positivo. Il valore assegnato a questo attributo non deve essere negativo. Se non si specifica l'attributo `retrycount`, viene utilizzato il valore predefinito zero.

L'attributo `retrywait` specifica il tempo di attesa, in secondi, prima di ritentare il richiamo del programma. Il valore assegnato a questo attributo non deve essere negativo. Se non si specifica l'attributo `retrywait`, viene utilizzato il valore predefinito zero.

L'attributo `successrc` è un'espressione utilizzata per stabilire quando il richiamo del programma viene eseguito correttamente. Il codice di ritorno del processo per il comando viene valutato utilizzando questa espressione. Il valore può essere composto da una o più espressioni combinate con un carattere barra verticale (|) per indicare OR booleano o una e commerciale (&) per indicare il valore booleano AND. Ogni espressione può essere uno dei seguenti tipi di espressione:

- Un numero che indica un test di uguaglianza tra il codice di ritorno del processo e il numero.
- Un numero prefissato con un carattere maggiore di (>) per indicare un test maggiore di tra il numero e il codice di ritorno del processo.
- Un numero preceduto da un carattere minore di (<) per indicare un test minore di quello tra il numero e il codice di ritorno del processo.
- Un numero preceduto da un carattere punto esclamativo (!) per indicare un test non uguale tra il numero e il codice di ritorno del processo. Ad esempio: `> 2 & < 7 &! 5 | 0 | 14` viene interpretato correttamente come i seguenti codici di ritorno: 0, 3, 4, 6, 14. Tutti gli altri codici di ritorno vengono interpretati come non riusciti.

Se non si specifica l'attributo `successrc`, viene utilizzato il valore predefinito zero. Ciò significa che si ritiene che il comando sia stato eseguito correttamente se, e solo se, restituisce un codice di zero.

Per uno script Ant, generalmente si specificano gli elementi `<target>` e `<property>`. I valori dell'elemento `<target>` devono corrispondere ai nomi di destinazione nello script Ant.

Per i programmi eseguibili, è possibile specificare elementi `<argument>`. Gli elementi argomento nidificati specificano gli argomenti da passare al programma richiamato come parte del richiamo del programma. Gli argomenti del programma sono creati dai valori specificati dagli elementi argomento nell'ordine in cui vengono rilevati gli elementi argomento. È possibile specificare zero o più elementi argomento come elementi nidificati di un richiamo del programma.

L'amministratore definisce e avvia il monitor normalmente utilizzando il documento XML di definizione attività che include l'elemento `<managedCall>`. Ad esempio:

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*.go
```

Il percorso del documento XML di definizioni di trasferimento deve essere sul filesystem locale da cui si esegue il comando **fteCreateMonitor** (in questo esempio `/tasks/cleanuptask.xml`). Il documento `cleanuptask.xml` viene utilizzato solo per creare il monitoraggio risorse. Tutte le attività a cui fa riferimento il documento `cleanuptask.xml` (script Ant o lavori JCL) devono trovarsi nel percorso del comando dell'agent di monitoraggio. Quando la condizione del trigger di monitoraggio viene soddisfatta, tutte le variabili nell'XML di definizione dell'attività vengono sostituite con i valori effettivi dal monitoraggio. Quindi, ad esempio `${FilePath}` viene sostituito nel messaggio di richiesta inviato all'agente

con `/monitored/cleanup.go`. Il messaggio di richiesta viene inserito sulla coda comandi dell'agente. Il processore comandi rileva che la richiesta è per una chiamata di programma e avvia il programma specificato. Se viene richiamato un comando di tipo `antscript`, viene avviata una nuova JVM e l'attività Ant viene eseguita nella nuova JVM. Per ulteriori informazioni sull'utilizzo della sostituzione delle variabili, consultare [Personalizzazione delle attività con la sostituzione delle variabili](#).

### Concetti correlati

[“Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile” a pagina 228](#)

Quando le condizioni del trigger di un controllo risorse attivo vengono soddisfatte, viene richiamata l'attività definita. Oltre a richiamare l'attività di trasferimento o comando con lo stesso agente di destinazione o lo stesso nome file di destinazione ogni volta, è anche possibile modificare la definizione dell'attività in fase di runtime. A tale scopo, inserire i nomi delle variabili nel file XML di definizione attività. Quando il controllo determina che le condizioni di trigger sono soddisfatte e che la definizione dell'attività contiene nomi di variabili, sostituisce i nomi di variabili con valori di variabile e richiama l'attività.

### Riferimenti correlati

[Formato messaggio di richiesta trasferimento file](#)  
[proprietà `commandPath` MFT](#)

### Creazione di un documento di definizione attività modificando un documento generato

È possibile creare il documento di definizioni di attività di monitoraggio modificando il documento XML generato dall'opzione `-gt` di `fteCreateTransfer`.

### Informazioni su questa attività

Il documento generato dispone di un elemento `<request>` seguito da `<managedTransfer>`. Per convertire questa definizione attività in una struttura `<managedCall>` valida, attenersi alla seguente procedura:

### Procedura

1. Sostituire le tag di inizio e fine `<managedTransfer>` con le tag `<managedCall>`.
2. Rimuovere tutti gli elementi `<schedule>` e i nodi secondari.
3. Sostituire le tag di inizio e di fine `<sourceAgent>` con `<agent>` per corrispondere ai dettagli di configurazione dell'agente di monitoraggio.
4. Rimuovere elementi `<destinationAgent>` e `<trigger>`.
5. Rimuovere `<item>` elementi.
6. Rimuovere gli elementi `preSourceCall`, `postSourceCall`, `preDestinationCall` e `postDestinationCall`.
7. Inserire una nuova struttura `<call>...</call>` all'interno dell'elemento `<transferSet>`. Questa struttura contiene la definizione del comando come mostrato nel seguente esempio:

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="{FileName}"/>
    <property name="trigger.path" value="{FilePath}"/>
  </command>
</call>
```

### Esempio

È inoltre possibile conservare l'elemento `<managedTransfer>` includendo tutti i dettagli di trasferimento file e inserire fino a quattro chiamate di comando. In questo caso, inserire una selezione dei seguenti elementi di chiamata tra gli elementi `<metaDataSet>` e `<item>`:

### **preSourceCall**

Richiamare un programma sull'agent di origine prima di avviare il trasferimento.

### **postSourceCall**

Richiamare un programma sull'agent di origine dopo aver completato il trasferimento.

### **preDestinationCall**

Richiamare un programma nell'agent di destinazione prima di avviare il trasferimento.

### **postDestinationCall**

Richiamare un programma nell'agent di destinazione dopo aver completato il trasferimento.

Ciascuno di questi elementi prende la struttura dell'elemento <command> come descritto nell'esempio precedente. Lo schema FileTransfer.xsd definisce i tipi utilizzati dai vari elementi di chiamata.

Il seguente esempio mostra preSourceCall, postSourceCall, preDestinationCall e postDestinationCall in un documento di definizione attività:

```
:
<transferSet priority="1">
  <metaDataSet>
    <metaData key="key1">value1</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
      type="executable">
      <argument>report1.pdf</argument>
      <argument>true</argument>
    </command>
  </preSourceCall>
  <postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
      type="jcl">
      <argument>argument</argument>
    </command>
  </postSourceCall>
  <preDestinationCall>
    <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
      type="antscript">
      <target>step1</target>
      <property name="name" value="value"/>
    </command>
  </preDestinationCall>
  <postDestinationCall>
    <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
  </postDestinationCall>
  <item checksumMethod="none" mode="binary">
:
```

È possibile combinare diversi tipi di comando nel trasferimento. Gli elementi argomento, destinazione e proprietà sono facoltativi.

## **Monitoraggio di una directory e utilizzo della sostituzione della variabile**

È possibile monitorare una directory utilizzando il comando **fteCreateMonitor**. Il valore di una variabile di sostituzione può essere sostituito nella definizione XML dell'attività e utilizzato per definire il comportamento del trasferimento.

### **Informazioni su questa attività**

In questo esempio, l'agent di origine è denominato AGENT\_HOP. La directory monitorata da AGENT\_HOP è denominata /test/monitored. L'agent esegue il polling della directory ogni 5 minuti.

Dopo che un file .zip è stato scritto nella directory, l'applicazione che scrive il file nella directory scrive un file trigger nella stessa directory. Il nome del file trigger è uguale al nome del file .zip, ma ha un'estensione file diversa. Ad esempio, dopo aver scritto il file file1.zip nella directory, il file file1.go viene scritto nella directory. Il controllo risorse monitora la directory per i file che corrispondono al modello \*.go, quindi utilizza la sostituzione della variabile per richiedere un trasferimento del file .zip associato.

## Procedura

1. Creare l'attività XML che definisce l'attività che il monitoraggio esegue quando viene attivato.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/${fileName}{token=1}{separator=.}.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/${fileName}{token=1}{separator=.}.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Le variabili sostituite con i valori associati al file trigger sono evidenziate in **grassetto**. Questa attività XML viene salvata nel file /home/USER1/task.xml

2. Creare un monitoraggio risorse per monitorare l'indirizzario /test/monitored.

Inoltrare il seguente comando:

```
fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*.go -pi 5 -pu minutes
```

3. Un utente o un programma scrive il file jump.zip nella directory /test/monitored, quindi scrive il file jump.go nella directory.
4. Il monitoraggio viene attivato dall'esistenza del file jump.go. L'agent sostituisce le informazioni sul file trigger nell'XML dell'attività.

Ciò determina la trasformazione dell'attività XML in:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/jump.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/jump.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

## Risultati

Viene eseguito il trasferimento definito dall'attività XML. Il file `jump.zip` viene letto dalla directory `/test/monitored` da `AGENT_HOP` e viene trasferito in un file denominato `/out/jump.zip` ubicato sul sistema su cui è in esecuzione `AGENT_SKIP`.

### Concetti correlati

[“Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile” a pagina 228](#)

Quando le condizioni del trigger di un controllo risorse attivo vengono soddisfatte, viene richiamata l'attività definita. Oltre a richiamare l'attività di trasferimento o comando con lo stesso agent di destinazione o lo stesso nome file di destinazione ogni volta, è anche possibile modificare la definizione dell'attività in fase di runtime. A tale scopo, inserire i nomi delle variabili nel file XML di definizione attività. Quando il controllo determina che le condizioni di trigger sono soddisfatte e che la definizione dell'attività contiene nomi di variabili, sostituisce i nomi di variabili con valori di variabile e richiama l'attività.

### Attività correlate

[“Configurazione delle attività di monitoraggio MFT per avviare comandi e script” a pagina 222](#)

I monitoraggi risorse non sono limitati all'esecuzione di trasferimenti file come attività associata. È anche possibile configurare il monitor per richiamare altri comandi dall'agent di monitoraggio, inclusi i programmi eseguibili, gli script Ant o i lavori JCL. Per richiamare i comandi, modificare l'XML di definizione dell'attività di monitoraggio in modo da includere uno o più elementi di comando con i parametri di chiamata del comando corrispondenti, ad esempio argomenti e proprietà.

### Riferimenti correlati

**fteCreateMonitor**: [crea un monitoraggio risorse MFT](#)

## Esempio: configurazione di una risorsa MFT

È possibile specificare una coda IBM MQ come la risorsa che deve essere monitorata da un monitoraggio risorse utilizzando il parametro `-mq` con il comando **fteCreateMonitor**.

### Informazioni su questa attività

In questo esempio, la risorsa da monitorare è la coda `MONITORED_QUEUE`. Questa coda deve essere sul gestore code dell'agent di monitoraggio, `QM_NEPTUNE`. La condizione per cui la coda è monitorata è la presenza di un gruppo completo di messaggi. L'attività da eseguire se la condizione è soddisfatta è definita nel file `task.xml`.

**Nota:** Non creare più di un controllo risorse per monitorare una singola coda. Se lo si fa, si verifica un comportamento imprevedibile.

### Procedura

Immettere il seguente comando:

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

Il controllo controlla la coda ogni cinque minuti per vedere se la condizione `completeGroups` è `true`. Se sulla coda sono presenti uno o più gruppi completi, il monitoraggio esegue l'attività definita nel file `task.xml` una volta per ciascun gruppo completo.

## Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile

Quando le condizioni del trigger di un controllo risorse attivo vengono soddisfatte, viene richiamata l'attività definita. Oltre a richiamare l'attività di trasferimento o comando con lo stesso agent di destinazione o lo stesso nome file di destinazione ogni volta, è anche possibile modificare la definizione dell'attività in fase di runtime. A tale scopo, inserire i nomi delle variabili nel file XML di definizione attività. Quando il controllo determina che le condizioni di trigger sono soddisfatte e che la definizione dell'attività contiene nomi di variabili, sostituisce i nomi di variabili con valori di variabile e richiama l'attività.



**Attenzione:** I nomi delle variabili non sono sensibili al maiuscolo / minuscolo.

Le variabili utilizzate per la sostituzione sono disponibili solo per le condizioni di trigger positivo. Solo le condizioni trigger match e fileSize causano la sostituzione delle variabili. Se viene utilizzata una condizione noMatch e sono presenti nomi di variabili di sostituzione nella definizione dell'attività, l'attività non viene richiamata e il controllo genera un codice di ritorno di 110 e un messaggio di errore BFGDM0060E.

## Se la risorsa monitorata è una coda

Il valore di qualsiasi proprietà del messaggio IBM MQ nel primo messaggio da leggere dalla coda monitorata può essere sostituito nella definizione XML dell'attività.

Le proprietà del messaggio definite dall'utente hanno come prefisso `usr.`, ma non includono questo prefisso nel nome della variabile. I nomi delle variabili devono essere preceduti dal simbolo del dollaro (\$) e racchiusi tra parentesi graffe {}.

Ad esempio, `${destFileName}` viene sostituito con il valore della proprietà del messaggio `usr.destFileName` del primo messaggio da leggere dalla coda di origine. Per ulteriori informazioni, consultare [MQ message properties read by MFT from messages on source queues](#) e [“Monitoraggio di una coda e utilizzo della sostituzione di variabili”](#) a pagina 234.

Se una variabile non è definita come una proprietà del messaggio, il monitoraggio riporta un errore BFGDM0060E e restituisce il codice di ritorno 110 (la sostituzione della variabile dell'attività di controllo non è riuscita). Inoltre, l'agent scrive il seguente messaggio di errore nel relativo log eventi (`outputN.log`):

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Se la registrazione del monitoraggio delle risorse moderata o verbose è abilitata per il monitoraggio, il monitoraggio scrive il seguente messaggio nel log degli eventi del monitoraggio delle risorse dell'agent (`resmoneventN.log`):

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Consultare [“Registrazione dei monitoraggi delle risorse MFT”](#) a pagina 239 per ulteriori informazioni sulla registrazione del monitoraggio delle risorse.

La seguente tabella mostra le variabili di sostituzione fornite per impostazione predefinita. Ad esempio, `${AGENTNAME}` viene sostituito con il nome dell'agent di monitoraggio risorse.

Variabile	Descrizione
NomeAgent	Il nome dell'agent del controllo risorse.
QUEUENAME	Il nome della coda monitorata.
Codifica	La codifica dei caratteri del primo messaggio della coda o del primo messaggio di un gruppo.
MessageId	L'ID del messaggio IBM MQ del primo messaggio sulla coda o del primo messaggio nel gruppo.
GroupID	L'ID gruppo IBM MQ del gruppo o l'ID messaggio se viene trovato solo un singolo messaggio. Questa variabile è impostata solo se si stanno controllando gruppi completi.
CurrentTime	Una data/ora basata sull'ora locale in cui è stato attivato il monitoraggio. Il valore data / ora è univoco per l'agent.

Tabella 10. Variabili di sostituzione fornite per impostazione predefinita (Continua)

Variabile	Descrizione
CurrentTimeStampUTC	Una data/ora basata sull'ora, nel fuso orario UTC, in cui è stato attivato il monitoraggio. Il valore data / ora è univoco per l'agent.

## Se la risorsa monitorata è una directory

La seguente tabella mostra la serie di nomi di variabili che possono essere sostituiti nella definizione XML dell'attività.

Tabella 11. Variabili che possono essere sostituite

Variabile	Descrizione
FilePath	Il percorso completo del file di trigger.
FileName	La parte nome file del trigger.
LastModified	L'ora di ultima modifica del file di trigger. Questa ora è espressa come l'ora locale del fuso orario in cui è in esecuzione l'agent ed è formattata come ora ISO 8601.
DataLastModified	La data di ultima modifica del file di trigger. Questa data è espressa come la data locale del fuso orario in cui è in esecuzione l'agent ed è formattata come data ISO 8601.
LastModifiedTimeUTC	L'ora di ultima modifica del file di trigger. Questa ora è espressa come l'ora locale convertita nel fuso orario UTC e viene formattata come ora ISO 8601
LastModifiedDateUTC	La data di ultima modifica del file di trigger. Tale data viene espressa come data locale convertita nel fuso orario UTC e ha il formato data ISO 8601.
AgentName	Il nome dell'agent del controllo risorse.
CurrentTime	Una data / ora basata sull'ora locale in cui è stato attivato il monitoraggio. Il valore data / ora è univoco per l'agent.
CurrentTimeStampUTC	Una data / ora basata sull'ora nel fuso orario UTC in cui è stato attivato il monitoraggio. Il valore data / ora è univoco per l'agent.

## Se la risorsa monitorata è un file trigger

La seguente tabella mostra la serie di nomi di variabile che possono essere sostituiti quando un controllo risorse utilizza il contenuto di un file trigger per determinare i file che devono essere trasferiti.

Tabella 12. Variabili che possono essere sostituite quando si utilizza un file trigger

Variabile	Descrizione
contentSource	Il nome percorso completo del file di origine.
contentDestination	Il nome percorso completo del file di destinazione.

I nomi delle variabili devono essere preceduti dal simbolo del dollaro (\$) e racchiusi tra parentesi graffe, {}. Ad esempio, \${FilePath} viene sostituito con il percorso file completo del file trigger corrispondente.

Ci sono due parole chiave speciali che possono essere applicate ai nomi delle variabili per fornire ulteriore rifinitura. Sono:

### token

L'indice token da sostituire (a partire da 1 da sinistra e a partire da -1 da destra)

## separatore

Un singolo carattere per suddividere in token il valore della variabile. Il valore predefinito è il carattere barra (/) su piattaforme AIX and Linux o il carattere barra rovesciata (\) su piattaforme Windows, ma il separatore può essere qualsiasi carattere valido che può essere presente nel valore della variabile.

Se la parola chiave separatore viene specificata in un nome di variabile, il valore della variabile viene suddiviso in token in base al carattere separatore.

Il valore assegnato alla parola chiave token viene utilizzato come indice per selezionare quale token utilizzare per sostituire il nome della variabile. L'indice token è relativo al primo carattere nella variabile e inizia da 1. Se la parola chiave del token non viene specificata, viene inserita l'intera variabile.

Tutti i valori sostituiti in un nome agent nell'XML del messaggio vengono trattati in modo non sensibile al maiuscolo / minuscolo. Tutti i nomi Managed File Transfer Agent sono in maiuscolo. Se il valore Paris viene sostituito in un attributo agent nell'XML del messaggio, questo valore viene interpretato come un riferimento all'agent PARIS.

## Concetti correlati

“Esempi: sostituzione di variabili per le definizioni di monitoraggio delle risorse” a pagina 231  
Esempi di sostituzione di variabili per definizioni di monitoraggio risorse utilizzando XML e IBM MQ Explorer.

## Attività correlate

[Cosa fare se la sostituzione della variabile fa sì che più file passino a un singolo nome file](#)

## Esempi: sostituzione di variabili per le definizioni di monitoraggio delle risorse

Esempi di sostituzione di variabili per definizioni di monitoraggio risorse utilizzando XML e IBM MQ Explorer.

## Esempi che mostrano il modo in cui funziona la sostituzione delle variabili

Supponendo che il percorso del file del trigger corrispondente sia c:\MONITOR\REPORTS\Paris\Report2009.doc su Windows e /MONITOR/REPORTS/Paris/Report2009.doc su piattaforme AIX and Linux, le variabili vengono sostituite come mostrato nella seguente tabella.

Specifica della variabile	Dopo la sostituzione della variabile
\${FilePath}	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux :/MONITOR/REPORTS/Paris/Report2009.doc
\${FilePath{token=1}{separator=.}}	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux :/MONITOR/REPORTS/Paris/Report2009
\${FilePath{token=2}{separator=.}}	Windows : doc AIX and Linux : doc
\${FilePath{token=3}}	WindowsReport AIX and Linux : Parigi

È inoltre possibile specificare un indice di token negativo per selezionare i token relativi all'ultimo carattere della variabile, come mostrato nella tabella seguente. Gli esempi nella tabella utilizzano lo stesso valore di variabile, c:\MONITOR\REPORTS\Paris\Report2009.doc su Windows e /MONITOR/REPORTS/Paris/Report2009.doc su AIX and Linux.

<i>Tabella 14. Esempi di utilizzo di un indice token negativo</i>	
<b>Specifica della variabile</b>	<b>Dopo la sostituzione della variabile</b>
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux :/MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux :/MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\}}</code>	Windows : Parigi AIX and Linux : Parigi
<code>\${FilePath{token=-4}}</code>	Windows : MONITOR AIX and Linux : MONITOR

Le variabili utilizzate per la sostituzione sono disponibili solo per le seguenti condizioni di trigger positivo e l'opzione noSizeChange, che è un'eccezione alla regola di condizione di trigger positivo:

- corrisponde
- fileSize
- noSizeModifica

Se viene utilizzata una condizione noMatch e sono presenti nomi di variabili di sostituzione nella definizione dell'attività, l'attività non viene richiamata e il controllo genera un codice di ritorno di 110 e un messaggio di errore BFGDM0060E.

## Esempio di utilizzo di XML

La seguente definizione di attività di esempio XML utilizza il nome dell'agent di monitoraggio come agent di origine per il trasferimento (Paris), utilizza la penultima directory nel percorso file come nome dell'agent di destinazione per il trasferimento (Report2009) e rinomina il file trasferito in modo che sia la root del nome file trigger con estensione .rpt.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="${AgentName}" QMgr="QM1" />
    <destinationAgent agent="${FilePath{token=-2}}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${FileName{token=1}{separator=.}}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

```
</managedTransfer>
</request>
```

Ciò determina la trasformazione dell'attività XML in:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="QM1" />
    <destinationAgent agent="Paris" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/Report2009.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

La variabile `${FilePath}{token=-2}` nell'attributo `agent` dell'elemento `<destinationAgent>` viene sostituita con il valore `Paris`. Questo valore viene trattato in modo non sensibile al maiuscolo / minuscolo e interpretato come un riferimento all'agent `PARIS`.

## Esempi che utilizzano IBM MQ Explorer

Quando si crea un monitoraggio risorse tramite IBM MQ Explorer una volta specificate le proprietà del monitoraggio e le condizioni di trigger, viene fornita l'opzione per aggiungere elementi di trasferimento al monitoraggio. I seguenti esempi dimostrano come è possibile utilizzare le variabili `${FilePath}` e `${FileName}` nel "**pannello Aggiungi un elemento di trasferimento**" per personalizzare i trasferimenti risultanti da una corrispondenza del monitoraggio risorse.

### Esempio 1

Per trasferire semplicemente il file di origine in un'altra posizione quando viene soddisfatta una condizione di trigger, è possibile utilizzare la variabile `${FilePath}` :

- Impostare il **Nome file** di origine su `${FilePath}`.
- Dal menu a discesa di **Tipo** per la destinazione, selezionare **Directory**.
- Impostare la destinazione **Nome file** in modo che sia l'ubicazione in cui si desidera trasferire il file di origine, ad esempio `C:\MFT\out\`.

### Esempio 2

Per trasferire il file di origine in un'altra ubicazione e modificare l'estensione del file, la variabile `${FileName}` può essere utilizzata insieme alla variabile `${FilePath}` :

Nel seguente esempio si presume che il percorso del file di origine sia uguale a `C:\MONITOR\REPORTS\Paris\Report2009.doc`:

- Impostare il **Nome file** di origine su `${FilePath}`.
- Impostare il **Nome file** di destinazione come ubicazione in cui si desidera trasferire il file di origine, seguito da `${FileName}{token=1}{separator=.}`, seguito dalla nuova estensione del file. Ad esempio, può essere `C:\MFT\out\${FileName}{token=1}{separator=.}.rpt`, che equivale a `C:\MFT\out\Report2009.rpt` con il nome file di origine.

### Esempio 3

Per utilizzare parte del percorso file del file di origine per determinare la destinazione del trasferimento, la variabile `${FilePath}` può essere utilizzata insieme alle specifiche del token e del separatore.

Nel seguente esempio si assume che il percorso file del file di origine sia uguale a `C:\MONITOR\REPORTS\Paris\Report2009.doc`.

È possibile utilizzare parte del percorso del file di origine per determinare la destinazione del file. Utilizzando l'esempio di percorso file di `C:\MONITOR\REPORTS\Paris\Report2009.doc`, se il file deve essere trasferito in una cartella in base all'ubicazione del file di origine, ovvero `Paris` in questo esempio, è possibile effettuare le seguenti operazioni:

- Impostare il **Nome file** di origine su `${FilePath}`.
- Impostare la destinazione **Nome file** in modo che sia la destinazione in cui si trovano le cartelle per ogni ubicazione, quindi accodare la parte di destinazione del percorso file e il nome file. Ad esempio, può essere `C:\MFT\out\${FilePath}token=-2}{separator=} }\${FileName}`, che equivale a `C:\MFT\out\Paris\Report2009.doc` con il nome file di origine.

### Concetti correlati

[“Personalizzazione delle attività di monitoraggio delle risorse MFT con la sostituzione della variabile” a pagina 228](#)

Quando le condizioni del trigger di un controllo risorse attivo vengono soddisfatte, viene richiamata l'attività definita. Oltre a richiamare l'attività di trasferimento o comando con lo stesso agent di destinazione o lo stesso nome file di destinazione ogni volta, è anche possibile modificare la definizione dell'attività in fase di runtime. A tale scopo, inserire i nomi delle variabili nel file XML di definizione attività. Quando il controllo determina che le condizioni di trigger sono soddisfatte e che la definizione dell'attività contiene nomi di variabili, sostituisce i nomi di variabili con valori di variabile e richiama l'attività.

### Attività correlate

[Cosa fare se la sostituzione della variabile fa sì che più file passino a un singolo nome file](#)

## Monitoraggio di una coda e utilizzo della sostituzione di variabili

È possibile monitorare una coda e trasferire messaggi dalla coda monitorata a un file utilizzando il comando `fteCreateMonitor`. Il valore di qualsiasi proprietà del messaggio IBM MQ nel primo messaggio da leggere dalla coda monitorata può essere sostituito nella definizione XML dell'attività e utilizzato per definire il funzionamento del trasferimento.

### Informazioni su questa attività

In questo esempio, l'agent di origine è denominato `AGENT_VENUS`, che si connette a `QM_VENUS`. La coda monitorata da `AGENT_VENUS` è denominata `START_QUEUE` e si trova su `QM_VENUS`. L'agent esegue il polling della coda ogni 30 minuti.

Quando un gruppo completo di messaggi viene scritto nella coda, l'attività di monitoraggio invia il gruppo di messaggi a uno dei diversi agent di destinazione, tutti connessi al gestore code `QM_MARS`. Il nome del file a cui viene trasferito il gruppo di messaggi viene definito dalla proprietà IBM MQ message `usr.fileName` sul primo messaggio del gruppo. Il nome dell'agent a cui viene inviato il gruppo di messaggi è definito dalla IBM MQ proprietà del messaggio `usr.toAgent` sul primo messaggio nel gruppo. Se l'intestazione `usr.toAgent` non è impostata, il valore predefinito da utilizzare per l'agent di destinazione è `AGENT_MAGENTA`.

Quando si specificano `useGroups="true"`, se non si specifica anche `groupId="${GROUPID}"`, il trasferimento acquisisce solo il primo messaggio sulla coda. Ad esempio, se si sta utilizzando la sostituzione della variabile per generare il `fileName`, è possibile che il contenuto di `a.txt` non sia corretto. Ciò è dovuto al fatto che `fileName` viene generato dal monitoraggio, ma il trasferimento in realtà riceve un messaggio che non è quello che dovrebbe generare il file denominato `fileName`.

### Procedura

1. Creare l'attività XML che definisce l'attività che il monitoraggio esegue quando viene attivato.

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="${toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

Le variabili sostituite con i valori delle intestazioni dei messaggi IBM MQ sono evidenziati in **grassetto**. Questa attività XML viene salvata nel file /home/USER1/task.xml

2. Creare un monitoraggio risorse per monitorare la coda START\_QUEUE.

Inoltrare il seguente comando:

```

fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA

```

3. Un utente o un programma scrive un gruppo di messaggi nella coda START\_QUEUE.

Il primo messaggio in questo gruppo ha le seguenti proprietà del messaggio IBM MQ :

```

usr.fileName=larmer
usr.toAgent=AGENT_VIOLET

```

4. Il controllo viene attivato quando viene scritto il gruppo completo. L'agent sostituisce le proprietà del messaggio IBM MQ nell'XML dell'attività.

Ciò determina la trasformazione dell'attività XML in:

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

## Risultati

Viene effettuato il trasferimento definito dall'attività XML. Il gruppo completo di messaggi letti da START\_QUEUE da AGENT\_VENUS viene scritto in un file denominato /reports/larmer.rpt sul sistema su cui è in esecuzione AGENT\_VIOLET.

## Operazioni successive

### Trasferimento di ogni messaggio in un file separato

Se si desidera monitorare una coda e trasferire ogni messaggio in un file separato, è possibile utilizzare una tecnica simile a quella descritta in precedenza in questo argomento.

1. Creare il monitoraggio come descritto in precedenza, specificando il parametro **-tr completeGroups** nel comando **fteCreateMonitor**.
2. Nell'XML dell'attività specificare quanto segue:

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

Tuttavia, quando si inserono i messaggi nella coda di origine, non inserirli in un gruppo IBM MQ. Aggiungere le proprietà del messaggio IBM MQ a ciascun messaggio. Ad esempio, specificare la proprietà `usr.filename` con un valore di nome file univoco per ciascun messaggio. Ciò fa sì che Managed File Transfer Agent consideri ciascun messaggio sulla coda di origine come un gruppo separato.

## Configurazione del comportamento dei tentativi di monitoraggio per i trasferimenti da messaggio a file

Se un trasferimento da messaggio a file attivato da un controllo risorse non riesce e lascia il gruppo di messaggi che ha attivato il controllo sulla coda, tale trasferimento viene reinoltrato a intervalli di polling successivi. Il numero di volte in cui il trasferimento viene reinoltrato è limitato dalla proprietà **monitorGroupRetryLimit** dell'agent di monitoraggio.

## Informazioni su questa attività

Ogni volta che viene attivato un nuovo trasferimento da messaggio a file, viene generato un nuovo ID trasferimento per l'attività di trasferimento.

Se l'agent viene riavviato, il monitoraggio attiva di nuovo un trasferimento anche se il numero di volte in cui il trasferimento è stato attivato ha superato il valore di **monitorGroupRetryLimit** nel file `agent.properties`. Il valore della proprietà **monitorGroupRetryLimit** è il numero massimo di volte in cui un monitoraggio attiva nuovamente un trasferimento da messaggio a file se il gruppo di messaggi esiste ancora nella coda. Il valore predefinito di questa proprietà è 10. Il valore di questa proprietà può essere impostato su qualsiasi valore intero positivo o -1. Se il valore -1 viene specificato per questa proprietà, il monitor attiva di nuovo il trasferimento un numero illimitato di volte, fino a quando la condizione trigger non viene soddisfatta.

Se un tentativo di trasferimento fa sì che il numero di volte in cui il trasferimento è stato attivato superi il valore di **monitorGroupRetryLimit**, l'agent scrive un errore nel relativo log eventi.

Un singolo messaggio viene considerato come se fosse un singolo gruppo e il trasferimento viene attivato nuovamente ad ogni intervallo di polling mentre il messaggio rimane sulla coda e mentre il numero di volte in cui il trasferimento è stato attivato è inferiore al valore di **monitorGroupRetryLimit**.

Per impostare la proprietà **monitorGroupRetryLimit** sull'agent di monitoraggio, effettuare le seguenti operazioni:

## Procedura

1. Arrestare l'agent di monitoraggio, utilizzando il comando **fteStopAgent**.
2. Modificare il file `agent.properties` per l'agent di monitoraggio per includere la seguente riga:

```
monitorGroupRetryLimit=number_of_retries
```

Il file `agent.properties` si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name`.

3. Avviare l'agent di monitoraggio utilizzando il comando **fteStartAgent** .

### Attività correlate

“Esempio: configurazione di una risorsa MFT” a pagina 228

È possibile specificare una coda IBM MQ come la risorsa che deve essere monitorata da un monitoraggio risorse utilizzando il parametro **-mq** con il comando **fteCreateMonitor** .

## Utilizzo di un file trigger

È possibile utilizzare il contenuto di un file trigger in un controllo risorse per definire una serie di file da trasferire in una richiesta di trasferimento singola. Ogni volta che viene rilevato un file trigger corrispondente, il relativo contenuto viene analizzato per i percorsi file di origine e facoltativamente per i percorsi file di destinazione. Questi percorsi file vengono quindi utilizzati per definire elementi file nel file XML di trasferimento attività specificato, inoltrato come singola richiesta di trasferimento all'agent. La definizione del monitoraggio risorse determina se il contenuto del trigger è abilitato.

È possibile abilitare l'attivazione del contenuto file quando si crea un controllo specificando il parametro **-tc** (contenuto trigger). Questo parametro **-tc** si applica solo alle opzioni trigger del file `match` e `noSizeChange`. Per ulteriori informazioni sulla creazione di un monitoraggio, consultare **fteCreateMonitor**: `create an MFT resource monitor`.

Quando si utilizza un file di contenuto trigger, il formato predefinito di ciascuna riga è:

- Un percorso file di origine singolo o
- Un percorso del file di origine e un percorso del file di destinazione, separati da una virgola

dove i caratteri spazio sono gestiti come parte dei percorsi file. È possibile modificare il formato di riga predefinito specificando i parametri **-tcr** e **-tcc** nel comando **fteCreateMonitor** . Per ulteriori informazioni, consultare “Opzioni avanzate” a pagina 238.

Una volta analizzato un file trigger, viene generato un elenco di percorsi file che viene applicato all'XML dell'attività di trasferimento specificato. Come con tutti i monitor, il formato dell'XML dell'attività di trasferimento è un XML dell'attività di trasferimento completo generato dal comando **fteCreateTransfer** con un singolo elemento o file definito. Il singolo elemento deve utilizzare le variabili di sostituzione `${contentSource}` e facoltativamente `${contentDestination}`, come sostituzioni per i percorsi dei file di origine e di destinazione. Il monitor espande l'XML dell'attività di trasferimento per includere un elemento file per ogni riga (percorso file) nel file trigger.

Non è possibile utilizzare il trigger del contenuto file con il parametro **-bs** perché il parametro **-tc** implica una richiesta di trasferimento per ogni file trigger.

## Esempio

Il seguente esempio definisce un monitor da attivare su un file che termina in `trig` e legge i percorsi file in tale file.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

Il comando **fteCreateTransfer** crea un file denominato `task.xml` per un singolo file con un percorso file di origine `${contentSource}`. Ad esempio:

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
```

```
</source>
</item>
```

Il comando **fteCreateMonitor** esegue la scansione dei file che terminano con `trig` nella directory `/home/trigdir` e utilizza i contenuti per creare una singola richiesta di trasferimento basata su `task.xml` per tutti i percorsi in tale file trigger. Il formato del file trigger deve essere un percorso file (solo origine) su ogni riga senza separatore virgola. Ad esempio:

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

Tutti i file vengono consegnati alla directory `/file/destdir` con il relativo nome file e non con il percorso file, ovvero `/home/file/first.txt` viene consegnato a `/file/destdir/first.txt`.

In alternativa, se si modifica il parametro **-dd /file/destdir** nel comando **fteCreateTransfer** in `-df ${contentDestination}` e il formato del contenuto di un file di trigger in *percorso file di origine, percorso file di destinazione*, è possibile definire percorsi di destinazione differenti per lo stesso agent di destinazione. Ad esempio:

```
/home/file/first.txt,/home/other/sixth.txt
```

L'ubicazione di destinazione diventa `/home/other/sixth.txt`.

Le variabili di sostituzione possono essere suddivise in token. Ad esempio, è possibile separare la parte del nome file dal percorso fornito utilizzando `${contentDestination{token=-1}}`. Pertanto, se la destinazione **fteCreateTransfer** è definita come `-df /file/destdir/${contentDestination{token=-1}}`, la nuova destinazione per `/home/file/first.txt` è `/file/destdir/sixth.txt`.

## Opzioni avanzate

È possibile cambiare il formato riga predefinito per il contenuto del file trigger utilizzando il parametro **-tcr regex**. Fornire un'espressione regolare che corrisponda al formato riga richiesto e fornire uno o due gruppi di cattura. Il primo gruppo di cattura è l'origine e il secondo, facoltativo, è la destinazione. Ad esempio:

- Il percorso di origine e di destinazione sono separati da un trattino:

```
((?:[^-]+)-((?:[^-]+)+)
```

In questo esempio, il separatore è definito in tre posizioni e tutte e tre le istanze del trattino, `-`, possono essere modificate in qualsiasi carattere. Assicurarsi di eseguire l'escape dei caratteri speciali.

- I percorsi di origine e di destinazione sono separati da una virgola con spazi finali. I commenti indicati da un cancelletto (`#`) vengono ignorati.

```
((?:[^\, ]+),((?:[^\, ]+)+) *(?:#.*)+
```

I percorsi file non possono contenere il simbolo del numero (`#`). Generalmente, una voce è la seguente: `/home/source/from.txt,/home/destination/to.txt # some comment`.

Se si utilizza il parametro **-tcr**, assicurarsi che l'espressione regolare sia ben progettata e verificata in modo che l'espressione possa rilevare gli errori e analizzare correttamente i file trigger.

È possibile invertire l'ordine della cattura utilizzando il parametro **-tcc destSrc**. Se si specifica questo parametro, il primo gruppo di cattura è il percorso del file di destinazione e il secondo gruppo è il percorso del file di origine.

## Come vengono gestiti gli errori

### File trigger vuoto

Se il file trigger è vuoto, il risultato non è un trasferimento file. Ossia, il controllo crea una richiesta di trasferimento, ma non viene specificato alcun elemento file.

### File trigger con errori

Se una voce in un file trigger non riesce ad analizzare rispetto al formato previsto, non viene generata alcuna richiesta di trasferimento. Viene pubblicato un log degli errori di monitoraggio e l'errore viene registrato anche nel log eventi. Il file trigger viene contrassegnato come elaborato e il monitoraggio non tenta di elaborare nuovamente il file fino a quando il file non viene aggiornato.

### XML attività di trasferimento non corrispondente

L'XML dell'attività di trasferimento deve corrispondere al file di trigger, ovvero se l'XML dell'attività di trasferimento ha `{contentSource}` e `{contentDestination}`, tutti i file di trigger per tale monitoraggio devono avere percorsi di file di origine e di destinazione e in modo simile per l'inverso. Nel primo caso, il monitoraggio riporta un errore di sostituzione di `{contentDestination}` se il file trigger fornisce solo il percorso del file di origine.

## Esempi

Il seguente esempio è un trigger di contenuto di base in cui il contenuto di un file trigger ha solo un percorso file di origine:

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

Il parametro **-tcr** definisce due gruppi di cattura di una sequenza di caratteri separati da un carattere spazio. Il parametro **-tcc destSrc** e l'opzione indicano che i gruppi di cattura devono essere elaborati come destinazione e quindi come origine.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df ${contentDestination} $
{contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
-tcr "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

## Registrazione dei monitoraggi delle risorse MFT

È possibile ottenere informazioni diagnostiche sui monitoraggi risorse utilizzando la registrazione.

### Informazioni su questa attività

È possibile utilizzare la registrazione per i monitoraggi delle risorse utilizzando il comando **fteSetAgentLogLevel** o il file `agent.properties` per controllare la registrazione del monitoraggio delle risorse.

Si noti che i punti di traccia esistenti sono ancora utilizzati per la cattura delle informazioni.

I log di monitoraggio delle risorse vengono scritti in un file denominato `resmoneventN.log`, dove *N* indica un numero; ad esempio, `resmonevent0.log`. I file di log eventi registrano diverse azioni che si verificano quando un monitor esegue il polling di una risorsa, ad esempio una directory o una coda.



**Attenzione:** Tutti i monitoraggi risorse di un agent scrivono nello stesso file di log.

Per un esempio di output di un file `resmoneventN.log`, consultare [Cosa fare se il controllo risorse dell'indirizzario MFT non sta attivando i file.](#)

La seguente tabella elenca il tipo di eventi che il controllo risorse scrive nel file di log. La terza colonna descrive il livello di log necessario per catturare ogni evento in cui il livello più basso è INFO e il più alto è VERBOSE.

Tenere presente che l'impostazione di un livello di log superiore, scrive anche eventi di livello inferiore. Ad esempio, l'impostazione del livello di log su MODERATO scrive anche gli eventi di livello INFO, ma non gli eventi di livello VERBOSE.

Numero	Evento	Livello di log	Descrizione
1	Monitor creato	INFORMAZION I	È stato creato un monitoraggio risorse.
2	Monitoraggio eliminato	INFORMAZION I	È stato eliminato un controllo risorse.
3	Monitoraggio arrestato	INFORMAZION I	Un controllo risorse è stato arrestato.
4	Monitoraggio avviato	INFORMAZION I	È stato avviato un controllo risorse.
5	Avvia sondaggio	INFORMAZION I	Un monitoraggio risorse ha avviato un nuovo ciclo di polling.
6	Fine polling	INFORMAZION I	Un ciclo di polling del controllo risorse è terminato.
7	Corrispondenza modello	VERBOSE	È stato trovato un file nell'indirizzario di controllo dei trigger o un messaggio in una coda che corrisponde al modello specificato.
8	Mancata corrispondenza modello	VERBOSE	È stato trovato un file non corrispondente nell'indirizzario di controllo del trigger o un messaggio in una coda che non corrisponde al modello specificato.
9	Richiesta di trasferimento	INFORMAZION I	Un trasferimento è stato avviato dal controllo risorse.
10	Directory troppo profonda	VERBOSE	L'indirizzario monitorato dal controllo risorse contiene più sottoindirizzari di cui eseguire il polling rispetto al numero specificato nella configurazione del controllo risorse.
11	File bloccato	MODERATO	Il file trigger monitorato dal monitoraggio risorse è bloccato da un'altra elaborazione.
12	Dimensione file piccola	MODERATO	Il file trigger è più piccolo della dimensione specificata nella configurazione del controllo risorse.
13	Dimensione file instabile	MODERATO	Il file trigger viene modificato più frequentemente del previsto dalla configurazione del controllo risorse.
14	Troppi polling	MODERATO	Un controllo risorse ha eseguito il polling di un file trigger instabile troppe volte.
15	Elementi corrispondenti	INFORMAZION I	Numero totale di file trigger trovati nell'indirizzario sottoposto a polling da un controllo risorse.
16	Elementi trasferimento	INFORMAZION I	Numero totale di elementi nella richiesta di trasferimento.
17	FDC generato	MODERATO	Un controllo risorse ha generato un'eccezione.

Numero	Evento	Livello di log	Descrizione
18	Richiesta di trasferimento	INFORMAZIONE	Richiesta di trasferimento inoltrata dal monitoraggio risorse.
19	Avvio monitor non riuscito	MODERATO	Non è stato possibile avviare un controllo risorse.
20	Cronologia cancellata	INFORMAZIONE	Le informazioni sulla cronologia del controllo sono state eliminate.
21	Cancellazione cronologia monitor non riuscita	INFORMAZIONE	Il tentativo di cancellare le informazioni cronologiche del controllo non è riuscito.
22	ID trasferimento	INFORMAZIONE	L'ID della richiesta di trasferimento è stato inoltrato dal monitoraggio.
23	Batch	INFORMAZIONE	Numero totale di richieste di trasferimento per gli elementi corrispondenti: <i>N</i> , dove <i>N</i> è un numero.

## Procedura

- Per utilizzare **fteSetAgentLogLevel** per attivare e disattivare la registrazione del monitoraggio delle risorse, consultare [fteSetAgentLogLevel](#) per una descrizione del parametro **logMonitor** ed esempi su come utilizzare le diverse opzioni.
- Per utilizzare il file `agent.properties` per controllare la registrazione del monitoraggio risorse, consultare [Il file MFT agent.properties](#) per una descrizione delle ulteriori proprietà che consentono di eseguire le seguenti attività di registrazione:
  - Attiva o disattiva la registrazione
  - Limita la dimensione di ciascun file di log
  - Limita il numero di log che i monitoraggi risorse possono generare

## Esempio

Le seguenti serie di messaggi di esempio verbose livello di registrazione per l'agent HA2, sul gestore code MFTDEMO:

```
<?xml version="1.0"?>
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
  <log:originator>
    <log:request>
      <log:hostname>192.168.7.1</log:hostname>
      <log:userid>johndoe</log:userid>
    </log:request>
  </log:originator>
  <log:endpoint agent="HA2" QMgr="MFTDEMO"/>
  <log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

## Riferimenti correlati

[Comando di livello fteSetAgentLog](#)

[Il file MFT agent.properties](#)

## V 9.2.2 Avvio di un monitoraggio risorse MFT

Da IBM MQ 9.2.2, è possibile avviare i monitoraggi delle risorse senza dover arrestare o riavviare un agent utilizzando il comando **fteStartMonitor**.

## Prima di iniziare

Se la gestione delle autorizzazioni utente è stata attivata impostando l'attributo **authorityChecking** su vero nel file `agent.properties`, è necessario disporre dell'autorizzazione Monitora o Operazioni di monitoraggio per avviare un monitoraggio delle risorse. Per ulteriori informazioni sulla gestione delle autorizzazioni utente, consultare [Limitazione delle autorizzazioni utente sulle MFT azioni dell'agent](#).

## Informazioni su questa attività

È possibile eseguire il comando **fteStartMonitor** da qualsiasi sistema in cui è installato il componente dei comandi Managed File Transfer, il che significa che è possibile avviare un monitoraggio delle risorse da qualsiasi ubicazione e non sono limitati al sistema in cui è in esecuzione l'agent proprietario del monitoraggio delle risorse. Per informazioni sui parametri obbligatori e facoltativi per questo comando, consultare [fteStartMonitor \(avvio di un MFT monitoraggio risorse\)](#).

## Procedura

- Per individuare lo stato di un agente prima o dopo l'esecuzione del comando **fteStartMonitor**, utilizzare il comando **fteListMonitors** con il parametro **-v** come mostrato nel seguente esempio:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Per avviare un monitoraggio risorse in un agent in esecuzione sulla stessa macchina, immettere il seguente comando **fteStartMonitor**:

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- Per avviare un monitoraggio risorse in un agente in esecuzione su una macchina diversa, immettere il comando **fteStartMonitor** nel modo seguente:

```
fteStartMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Se il gestore code comandi è anche il gestore code agent per l'agent di monitoraggio, il parametro **-mm** è facoltativo, altrimenti è necessario specificare il gestore code agent con il parametro **-mm**.

## Risultati

Se l'agent è in esecuzione, il monitoraggio risorse viene avviato se è attualmente arrestato. Il comando emette i seguenti messaggi e registra un evento nel `output0.log` dell'agent.

```
BFGCL0816I: È stata emessa una richiesta per avviare il monitoraggio delle risorse  
'nome_monitoraggio' dell'agent 'nome_agent'.  
BFGCL0251I: La richiesta è stata completata correttamente.
```

Per informazioni sui messaggi emessi dal comando se non è in grado di avviare il monitoraggio delle risorse, consultare [fteStartMonitor \(avvio di un MFT monitoraggio delle risorse\)](#).

## Concetti correlati

[“Concetti di monitoraggio delle risorse MFT” a pagina 217](#)

Una panoramica dei concetti chiave della funzione di monitoraggio delle risorse Managed File Transfer.

## Attività correlate

[“Arresto di un monitoraggio risorse MFT” a pagina 242](#)

Da IBM MQ 9.2.2, è possibile arrestare i monitoraggi delle risorse senza dover arrestare o riavviare un agent utilizzando il comando **fteStopMonitor**.

## Riferimenti correlati

[fteStartMonitor \(avvia un controllo risorse MFT\)](#)

## V 9.2.2 Arresto di un monitoraggio risorse MFT

Da IBM MQ 9.2.2, è possibile arrestare i monitoraggi delle risorse senza dover arrestare o riavviare un agent utilizzando il comando **fteStopMonitor**.

## Prima di iniziare

Se la gestione dell'autorizzazione utente è stata abilitata impostando l'attributo **authorityChecking** su `true` nel file `agent.properties`, è necessario disporre dell'autorizzazione Monitora o Operazioni di monitoraggio per arrestare un monitoraggio delle risorse. Per ulteriori informazioni sulla gestione delle autorizzazioni utente, consultare [Limitazione delle autorizzazioni utente sulle MFT azioni dell'agent](#).

## Informazioni su questa attività

È possibile eseguire il comando **fteStopMonitor** da qualsiasi sistema in cui è installato il componente dei comandi Managed File Transfer, il che significa che è possibile arrestare un monitoraggio delle risorse da qualsiasi punto e non sono limitati al sistema in cui è in esecuzione l'agent proprietario del monitoraggio delle risorse. Per informazioni sui parametri richiesti e facoltativi per questo comando, consultare [fteStopMonitor \(stop an MFT resource monitor\)](#).

Quando un monitoraggio delle risorse viene arrestato, scrive un messaggio nel log eventi del monitoraggio delle risorse dell'agent, `resmoneventnumber.log`. Se il monitoraggio risorse viene arrestato con il comando **fteStopMonitor**, il messaggio include il nome dell'utente che ha emesso la richiesta di arresto:

```
Monitoraggio risorse arrestato dall'utente '< mquser_id>'
```

Un monitoraggio delle risorse viene avviato automaticamente se il relativo agent viene riavviato, anche se il monitoraggio delle risorse è stato precedentemente arrestato utilizzando il comando **fteStopMonitor**.

Gli agent elaborano le richieste di monitoraggio in modo seriale piuttosto che in parallelo, quindi, ad esempio, se un agent riceve una richiesta di arresto del monitoraggio M1 e quindi un'altra richiesta di arresto del monitoraggio M2 in rapida successione, arresta M1 prima di tentare di arrestare M2.

## Procedura

- Per individuare lo stato di un agente prima o dopo l'esecuzione del comando **fteStopMonitor**, utilizzare il comando **fteListMonitors** con il parametro **-v** come mostrato nel seguente esempio:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Per arrestare un monitoraggio risorse in un agente in esecuzione sulla stessa macchina, immettere il seguente comando **fteStopMonitor**:

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- Per arrestare un monitoraggio risorse in un agent in esecuzione su una macchina differente, immettere il comando **fteStopMonitor** nel modo seguente:

```
fteStopMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Se il gestore code comandi è anche il gestore code agent per l'agent di monitoraggio, il parametro **-mm** è facoltativo, altrimenti è necessario specificare il gestore code agent con il parametro **-mm**.

## Risultati

Se l'agent è in esecuzione, il monitoraggio delle risorse viene arrestato se è attualmente avviato. Il comando emette i seguenti messaggi e registra un evento nel `output0.log` dell'agent.

```
BFGCL0813I: È stata emessa una richiesta di arresto del monitoraggio risorse 'MNTR' dell'agent 'SOURCE'.
```

```
BFGCL0251I: La richiesta è stata completata correttamente.
```

Per informazioni sui messaggi emessi dal comando se non è in grado di arrestare il monitoraggio risorse, consultare [fteStopMonitor \(stop an MFT resource monitor\)](#)

## Concetti correlati

[“Concetti di monitoraggio delle risorse MFT” a pagina 217](#)

Una panoramica dei concetti chiave della funzione di monitoraggio delle risorse Managed File Transfer .

### Attività correlate

“Avvio di un monitoraggio risorse MFT” a pagina 241

Da IBM MQ 9.2.2, è possibile avviare i monitoraggi delle risorse senza dover arrestare o riavviare un agent utilizzando il comando **fteStartMonitor** .

### Riferimenti correlati

[fteStopMonitor](#) (arresta un controllo risorse MFT)

## Backup e ripristino dei monitoraggi delle risorse MFT

È possibile eseguire il backup dei monitoraggi risorse che si desidera rendere disponibili per un utilizzo futuro esportandone le definizioni in un file XML che è possibile importare per creare un nuovo monitoraggio risorse dal backup.

### Informazioni su questa attività

Potrebbe essere necessario eseguire il backup dei monitoraggi risorse precedentemente definiti in modo da poterne riutilizzare le definizioni in futuro, ad esempio per ricreare i monitoraggi risorse in un'infrastruttura differente o se un monitoraggio risorse deve essere ricreato a causa di problemi del gestore code.

È possibile eseguire il backup di una singola definizione del gestore risorse utilizzando il comando **fteCreateMonitor** o il comando **fteListMonitors** con il parametro **-ox** . In entrambi i casi, viene eseguito il backup della definizione del gestore risorse esportandola in un file XML. È quindi possibile utilizzare il parametro **-ix** del comando **fteCreateMonitor** per creare un nuovo gestore risorse importando la definizione dal file XML.

Con il parametro **-ox** , è possibile eseguire il backup di una sola definizione di monitoraggio risorse alla volta.

Da IBM MQ 9.1, il parametro **-od** viene aggiunto al comando **fteListMonitors** . Specificando questo parametro, è possibile eseguire il backup di più di un controllo risorse alla volta esportando le relative definizioni in massa in una directory specificata. Ogni definizione di monitoraggio delle risorse viene salvata in un file XML separato con un nome nel formato *agent name.monitor name.xml*.

Il parametro **-od** è particolarmente utile se si dispone di un numero elevato di monitoraggi delle risorse di cui si desidera eseguire il backup poiché è necessario eseguire il comando **fteListMonitors -od** una sola volta, invece di dover eseguire il comando **fteListMonitors -ox** separatamente per ciascuna definizione di risorsa o utilizzare uno script separato per eseguire il comando **fteListMonitors -ox** per ciascun monitoraggio delle risorse.

### Procedura

- Per eseguire il back up della definizione di un controllo risorse esportandolo in un file XML, utilizzare uno dei seguenti comandi:
  - Il comando **fteCreateMonitor** con il parametro **-ox** .
  - Il comando **fteListMonitors** con il parametro **-ox** .

Quando si sta utilizzando il parametro **-ox** , è necessario specificare anche i parametri **-ma** e **-mn** , come mostrato nel seguente esempio:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

- Per eseguire il backup di più definizioni di monitoraggio risorse esportandole in file XML in una directory specificata, utilizzare il comando **fteListMonitors** con il parametro **-od** come mostrato nel seguente esempio:

```
fteListMonitors -od /usr/mft/resmonbackup
```

È necessario specificare una directory di destinazione valida quando si esegue il backup dei monitoraggi risorse in massa. Non specificando un percorso di destinazione si genera un messaggio di errore come mostrato nel seguente esempio:

```
BFGCL0762E: Directory output non specificata. Eseguire di nuovo il comando specificando un percorso valido.
```

Il parametro **-od** non deve essere combinato con il parametro **-ox**, altrimenti viene visualizzato il seguente messaggio di errore:

```
BFGCL0761E: non è valido specificare insieme i parametri '- od' e '- ox'.
```

È possibile definire una particolare serie di monitoraggi risorse da includere nel backup. Ad esempio, utilizzando il parametro **-ma** per specificare il nome di un agent, è possibile eseguire il backup di tutti i monitoraggi risorse per tale agent, come mostrato nel seguente esempio:

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

È anche possibile utilizzare la corrispondenza con caratteri jolly includendo un carattere asterisco (\*) quando si definisce un pattern da utilizzare per la corrispondenza dei nomi agent o dei nomi di monitoraggio, o entrambi. Il seguente esempio esegue il backup di tutti i monitoraggi delle risorse che hanno nomi che corrispondono a un pattern specificato e che si trovano in un agent con un nome che corrisponde a un pattern specificato:

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

Mentre il comando è in esecuzione, vengono visualizzati i seguenti messaggi di report di avanzamento:

```
È stato rilevato un totale di numero definizioni di monitoraggio risorse corrispondenti.  
indice di numero definizioni di monitoraggio risorse salvate nel filesystem.
```

Se si sta utilizzando l'opzione verbose, il totale parziale è ancora visualizzato, ma invece di visualizzarlo *indice* di *numero* definizioni di monitoraggio risorse salvate nel filesystem

il comando visualizza il nome della definizione di monitor che si sta salvando, ad esempio:

```
BFGCL0762I: Definizione del monitoraggio 'FILEMON' dell'agent 'XFERAGENT' salvata come FILEMON.XFERAGENT.XML nel filesystem.
```

- Per eseguire il backup di un monitoraggio risorse per un determinato agente esportandolo in un file XML in una directory specificata, utilizzare il comando **fteListMonitors** con il parametro **-od**:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

L'uso del parametro **-od** per eseguire il backup di un singolo monitoraggio delle risorse è simile all'utilizzo del parametro **-ox**, tranne che il nome del file di output è nel formato *agent name.monitor name.xml*.

- Per ripristinare le definizioni di monitoraggio risorse da un backup, utilizzare il comando **fteCreateMonitor** con il parametro **-ix** come mostrato nel seguente esempio:

```
fteCreateMonitor -ix file name
```

Per ulteriori esempi su come utilizzare il parametro **-od**, consultare [fteListMonitors: list MFT resource monitors](#).

## Riferimenti correlati

**fteCreateMonitor**: crea un monitoraggio risorse MFT

[fteListMonitoraggi](#): elenco MFT monitoraggi risorse

## V 9.2.0 Cancellazione della cronologia del controllo risorse

È possibile cancellare la cronologia di un monitoraggio risorse in modo che un'altra richiesta di trasferimento file possa essere inoltrata per un file che non è stato trasferito in precedenza a causa di un errore. Per cancellare la cronologia del monitoraggio risorse, è possibile utilizzare il comando **fteClearMonitorHistory** o IBM MQ Explorer.

## Prima di iniziare

Se la gestione delle autorizzazioni utente è stata abilitata impostando l'attributo **authorityChecking** su `true` nel file `agent.properties`, l'utente che cancella la cronologia di monitoraggio deve disporre dell'autorizzazione appropriata, come mostrato nella seguente tabella.

Tabella 15. Autorizzazione utente richiesta per eseguire il comando **fteClearMonitorHistory**

Utente che cancella la cronologia del monitoraggio	Autorizzazione di accesso MFT	Autorizzazione richiesta
Lo stesso utente di quello che ha creato il monitoraggio risorse.	Monitor	SFOGLIA su SYSTEM.FTE.AUTHMON1. <nome_agent_monitoraggio>  Si tratta della stessa autorizzazione richiesta per creare o cancellare il controllo risorse.
Qualsiasi utente diverso dall'utente che ha creato il monitoraggio risorse.	Operazioni di monitoraggio	SET su SYSTEM.FTE.AUTHPS1. <nome_agent>  Si tratta della stessa autorizzazione richiesta per cancellare il controllo risorse.

Per ulteriori informazioni sulla gestione delle autorizzazioni utente, consultare [Limitazione delle autorizzazioni utente sulle MFT azioni dell'agent](#).

Se un utente senza l'autorizzazione richiesta tenta di cancellare la cronologia del monitoraggio delle risorse, il comando **fteClearMonitorHistory** emette un messaggio di errore e registra l'errore nel file `output0.log` dell'agent. Per ulteriori informazioni, consultare [fteClearMonitorHistory: clear resource monitor history](#).

## Informazioni su questa attività

Se un trasferimento file è stato avviato e un file non può essere trasferito per qualsiasi ragione, il controllo risorse non seleziona nuovamente questo file per il trasferimento nel successivo polling poiché la cronologia del monitoraggio indica che il file è stato visualizzato in un polling precedente e non è stato modificato da allora (consultare [“Concetti di monitoraggio delle risorse MFT”](#) a pagina 217).

Prima di IBM MQ 9.1.3, se un file non riesce a trasferire, il trasferimento file può essere avviato di nuovo solo se il file viene eliminato e quindi nuovamente inserito nella directory, se il file viene aggiornato in modo che l'attributo della data dell'ultima modifica venga modificato o se il monitoraggio risorse stesso viene ricreato.

Tuttavia, da IBM MQ 9.1.3, è possibile cancellare la cronologia del monitoraggio risorse utilizzando il comando **fteClearMonitorHistory** o utilizzando IBM MQ Explorer. La cancellazione della cronologia consente ad un'altra richiesta di trasferimento per un file che non è riuscito a trasferire di essere inoltrato senza la necessità di eliminare il file e quindi riposizionarlo nella directory oppure aggiornare il file per modificare l'attributo della data dell'ultima modifica, che è utile, ad esempio, in situazioni in cui è necessario trasferire il file ma non è possibile modificare il file. Essere in grado di cancellare la cronologia di un controllo risorse significa anche che non è necessario ricreare il controllo risorse per inoltrare un'altra richiesta di trasferimento per un file che non è riuscito a trasferire.

 Il membro  SCSQFCMD di esempio fornito con Managed File Transfer on z/OS include uno script JCL per cancellare la cronologia del controllo.

## Procedura

- Per utilizzare il comandi **fteClearMonitorHistory** per cancellare la cronologia del controllo risorse, immettere il comando nel formato seguente:

```
fteClearMonitorHistory -p <configuration> -ma <agent name> -mn <monitor name> -w 1000
```

Sono richiesti solo i parametri **-ma** e **-mn** . Tutti gli altri parametri sono facoltativi. Per ulteriori informazioni su come utilizzare il comando **fteClearMonitorHistory** , inclusi gli esempi, consultare [fteClearMonitorHistory: clear resource monitor history](#).

Se la cronologia viene cancellata correttamente, il comando visualizza il seguente messaggio:

```
BFGCL0780I: È stata emessa una richiesta di cancellazione della cronologia del monitoraggio delle risorse 'nome monitoraggio' dell'agent 'nome agent'.  
BFGCL0251I: La richiesta è stata completata correttamente.
```

e registra la riuscita nel file output0.log dell'agent.

Se il tentativo di cancellare la cronologia del monitoraggio risorse ha esito negativo, **fteClearMonitorHistory** emette un messaggio di errore e registra l'errore nel file output0.log dell'agent.

- Per utilizzare la vista Monitoraggio risorse nel plug-in IBM MQ Explorer MFT per cancellare la cronologia del monitoraggio risorse, fare clic con il tasto destro del mouse sul monitoraggio risorse e selezionare **Cancella cronologia** dal menu a discesa.

Se la cronologia viene cancellata correttamente, viene visualizzato il seguente messaggio:

```
BFGUI00171: La cronologia del monitoraggio risorse è stata cancellata correttamente.
```

Se il tentativo di cancellare la cronologia ha esito negativo, viene visualizzato un messaggio di errore. Ad esempio:

```
BFGUI0016E Impossibile cancellare la cronologia del monitoraggio risorse specificato - motivo 2059
```

## Utilizzo dei modelli di trasferimento file

È possibile utilizzare i modelli di trasferimento file per memorizzare le impostazioni di trasferimento file comuni per trasferimenti ripetuti o complessi. Creare un modello di trasferimento dalla riga comandi utilizzando il comando **fteCreateTemplate** oppure utilizzare IBM MQ Explorer per creare un modello di trasferimento utilizzando la procedura guidata **Crea nuovo modello per Managed File Transfer** oppure salvare un modello durante la creazione di un trasferimento file selezionando la casella di controllo **Salva impostazioni di trasferimento come modello** . La finestra **Modelli di trasferimento** visualizza tutti i modelli di trasferimento creati nella rete Managed File Transfer .

### Informazioni su questa attività

Per creare un modello di trasferimento dalla riga comandi, utilizzare il comando [fteCreateTemplate](#) . Quindi, quando si desidera inoltrare un modello di trasferimento creato sulla riga di comando, fare clic su **Inoltra** in IBM MQ Explorer.

Per visualizzare i modelli di trasferimento in IBM MQ Explorer, utilizzare la seguente procedura:

## Procedura

1. Espandere **Managed File Transfer** nella vista Navigator . **Managed File Transfer Central** viene visualizzato nella vista Contenuto.
2. Tutti i gestori code di coordinamento sono elencati nella vista Navigator . Espandere il nome del gestore code di coordinamento utilizzato per il trasferimento pianificato. Se si desidera modificare il gestore code di coordinamento a cui si è connessi, fare clic con il tasto destro del mouse sul nome del gestore code di coordinamento che si desidera utilizzare nella vista Navigator e fare clic su **Connetti**.
3. Fare clic su **Trasferisci modelli**. Viene visualizzata la finestra **Modelli di trasferimento** nella vista Contenuto.

4. La finestra **Modelli di trasferimento** elenca i seguenti dettagli sui trasferimenti file:

- a) **Nome** Il nome del modello di trasferimento file.
- b) **Origine** Il nome dell'agente utilizzato per trasferire il file dal sistema di origine.
- c) **File di origine** Il nome del file da trasferire sul sistema host.  
Espandere le informazioni del modello di trasferimento per visualizzare questo campo.
- d) **Destinazione** Il nome dell'agent utilizzato per ricevere il file sul sistema di destinazione.
- e) **File di destinazione** Il nome del file dopo il trasferimento sul sistema di destinazione.  
Espandere le informazioni del modello di trasferimento per visualizzare questo campo.
- f) **Inizio pianificato (fuso orario selezionato)** L'ora e la data in cui è pianificato l'avvio del trasferimento file nel fuso orario utilizzato dall'amministratore. Per modificare il fuso orario visualizzato, selezionare **Finestra > Preferenze > IBM MQ Explorer > Managed File Transfer** e selezionare un fuso orario alternativo dall'elenco **Fuso orario:** . Fare clic su **OK**.
- g) **Eventi trigger** Il tipo di evento che attiva l'avvio del trasferimento file. Il tipo può essere uno dei seguenti valori: esiste, non esiste, supera.

## Risultati

Per aggiornare quanto visualizzato nella finestra **Modelli di trasferimento** , fare clic sul pulsante **Aggiorna**  sulla barra degli strumenti della vista Contenuto.

Per inoltrare un modello di trasferimento e avviare il trasferimento definito nel modello, fare clic con il tasto destro del mouse sul nome del modello e fare clic su **Inoltra**.

Per cambiare un modello di trasferimento, fare clic con il pulsante destro del mouse sul nome del modello e selezionare **Modifica**. Tutti i file inclusi nel modello originale vengono elencati come parte di un gruppo di trasferimenti, anche se non sono stati inclusi come parte di un gruppo nel modello originale. Se si desidera rimuovere un file dal modello, è necessario selezionare la specifica file dal gruppo e fare clic su **Rimuovi selezionati**. Se si desidera aggiungere nuove specifiche file al modello, utilizzare i campi nel pannello del modello e fare clic su **Aggiungi al gruppo** . Dopo aver apportato le modifiche, viene richiesto di fornire un nuovo nome al template modificato.

Per creare un trasferimento file da un modello di trasferimento, fare clic con il tasto destro del mouse sul nome del modello e selezionare **Modifica come nuovo trasferimento**.

Per creare una copia duplicata di un modello di trasferimento, fare clic con il tasto destro del mouse sul nome del modello e selezionare **Duplica**. Il modello di trasferimento duplicato viene salvato automaticamente con lo stesso nome del modello originale, aggiunto con "(copia)".

Per eliminare un modello di trasferimento, fare clic con il tasto destro del mouse sul nome del modello e selezionare **Elimina**.

### Attività correlate

[“Creazione di un modello di trasferimento file utilizzando IBM MQ Explorer” a pagina 248](#)

È possibile creare un modello di trasferimento file da IBM MQ Explorer o dalla riga comandi. È quindi possibile utilizzare tale modello per creare nuovi trasferimenti di file utilizzando i dettagli del modello o inoltrare il modello per avviare il trasferimento file.

### Riferimenti correlati

[f>CreateTemplate](#): crea un nuovo modello di trasferimento file

[Modelli fteList](#)

[Modelli fteDelete](#)

## Creazione di un modello di trasferimento file utilizzando IBM MQ Explorer

È possibile creare un modello di trasferimento file da IBM MQ Explorer o dalla riga comandi. È quindi possibile utilizzare tale modello per creare nuovi trasferimenti di file utilizzando i dettagli del modello o inoltrare il modello per avviare il trasferimento file.

## Informazioni su questa attività

Per creare un modello di trasferimento file dalla riga comandi, utilizzare il comando [fteCreateTemplate](#) .

Per creare un template di trasferimento file utilizzando la procedura guidata **Crea nuovo template per Managed File Transfer** in IBM MQ Explorer, effettuare le seguenti operazioni:

### Procedura

1. Nella vista Navigator , fare clic su **Managed File Transfer. Managed File Transfer Central** viene visualizzato nella vista Contenuto.
2. Tutti i gestori code di coordinamento vengono visualizzati nella vista Navigator . Espandere il nome del gestore code di coordinamento utilizzato per il trasferimento pianificato. Se si desidera modificare il gestore code di coordinamento a cui si è connessi, fare clic con il tasto destro del mouse sul nome del gestore code di coordinamento che si desidera utilizzare nella vista Navigator e fare clic su **Connetti**.
3. Avviare la procedura guidata **Crea nuovo template per Managed File Transfer** facendo clic con il pulsante destro del mouse su **Template di trasferimento** e facendo clic su **Nuovo template**.
4. Seguire le istruzioni sui pannelli della procedura guidata. Per ogni pannello viene fornita una guida sensibile al contesto. Per accedere alla guida sensibile al contesto su Windows, premere F1. Su Linux, premere Ctrl+F1 o Shift+F1.

Se è stato creato un modello che contiene tutti i dettagli di trasferimento richiesti, assicurarsi di selezionare la casella di spunta **Salva impostazioni di trasferimento come modello** nella pagina **Riepilogo trasferimento** se questa casella di spunta non è già selezionata. Immettere anche un nome per il modello nel campo Nome. Se si crea un modello che non contiene ancora tutti i dettagli di trasferimento richiesti, la casella di spunta **Salva impostazioni di trasferimento come modello** viene selezionata automaticamente.

### Attività correlate

[“Utilizzo dei modelli di trasferimento file” a pagina 247](#)

È possibile utilizzare i modelli di trasferimento file per memorizzare le impostazioni di trasferimento file comuni per trasferimenti ripetuti o complessi. Creare un modello di trasferimento dalla riga comandi utilizzando il comando **fteCreateTemplate** oppure utilizzare IBM MQ Explorer per creare un modello di trasferimento utilizzando la procedura guidata **Crea nuovo modello per Managed File Transfer** oppure salvare un modello durante la creazione di un trasferimento file selezionando la casella di controllo **Salva impostazioni di trasferimento come modello** . La finestra **Modelli di trasferimento** visualizza tutti i modelli di trasferimento creati nella rete Managed File Transfer .

### Riferimenti correlati

[fteCreateTemplate](#): crea un nuovo modello di trasferimento file

[Modelli fteList](#)

[Modelli fteDelete](#)

## Backup di una definizione della maschera di trasferimento file

I modelli di trasferimento file contengono un documento XML che definisce le specifiche file di origine e di destinazione per il trasferimento. È possibile utilizzare questo file XML come input per il comando **fteCreateTemplate** per creare nuovamente un modello di trasferimento file.

## Informazioni su questa attività

Per eseguire il backup del documento XML contenente le specifiche file di origine e di destinazione per un modello di trasferimento, utilizzare il comando [fteCreateTransfer command](#) o IBM MQ Explorer. Per creare un file di backup formattato XML del modello di trasferimento, utilizzare la seguente procedura:

## Procedura

- Metodo uno: utilizzare il parametro **-gt** su un comando [fteCreateTransfer](#) per generare un messaggio XML del template di trasferimento in un nuovo file.
- Metodo due: creare il modello utilizzando IBM MQ Explorer.  
Quando si arriva alla pagina *Riepilogo modello di trasferimento* :
  - a) Copiare l' *Anteprima XML del messaggio di richiesta*.
  - b) Salvare questo messaggio XML modello di trasferimento in un nuovo file.
- Metodo tre: utilizzare IBM MQ Explorer per eseguire il backup dei modelli esistenti.
  - a) Andare a **Managed File Transfer > Nome gestore code > Template di trasferimento**.
  - b) Nel pannello Trasferisci, evidenziare il modello di cui è necessario eseguire il backup, fare clic con il pulsante destro del mouse e selezionare **Modifica** dal menu a comparsa.
  - c) Fare clic su **Next** fino a quando non si arriva alla pagina *Transfer template summary* .
  - d) Copiare l' *Anteprima XML del messaggio di richiesta*.
  - e) Salvare questo messaggio XML modello di trasferimento in un nuovo file.

## Risultati

È possibile utilizzare il file di messaggi XML del modello di trasferimento, creato da uno dei metodi precedenti, come input per il comando [fteCreateTemplate](#) . Per i dettagli su come utilizzare questo comando, fare riferimento al comando **fteCreateTemplate** .

### Riferimenti correlati

[Comando fteCreateTemplate](#)

[comando fteListTemplates](#)

## Trasferimento di dati dai file ai messaggi

È possibile utilizzare la funzione file - to - message di Managed File Transfer per trasferire i dati da un file a un singolo messaggio o a più messaggi su una coda IBM MQ .

Per eseguire trasferimenti da file a messaggio e da messaggio a file, sia l'agent di origine che quello di destinazione del trasferimento devono essere all'indirizzo IBM WebSphere MQ 7.5 o successivo oppure all'indirizzo IBM WebSphere MQ File Transfer Edition 7.0.3o successivo. Per informazioni sui trasferimenti da messaggio a file, consultare [“Trasferimento dei dati dai messaggi ai file”](#) a pagina 258.

L'agent di destinazione per un trasferimento file - to - message non può essere un agent bridge di protocollo o un agent bridge Connect:Direct .

È possibile trasferire i dati del file ai dati del messaggio IBM MQ . I messaggi IBM MQ possono essere letti e utilizzati dalle applicazioni. Sono supportati i seguenti tipi di trasferimento file - to - message:

- Da un singolo file a un singolo messaggio. Il messaggio non dispone di un ID gruppo IBM MQ impostato.
- Da un singolo file a più messaggi, suddividendo il file in messaggi di una data lunghezza. I messaggi hanno tutti lo stesso ID gruppo IBM MQ .
- Da un singolo file a più messaggi, suddividendo un file di testo in un delimitatore di espressione regolare Java . I messaggi hanno tutti lo stesso ID gruppo IBM MQ .
- Da un singolo file a più messaggi, suddividendo un file binario con un delimitatore esadecimale. I messaggi hanno tutti lo stesso ID gruppo IBM MQ .

Se si desidera suddividere un file binario utilizzando una sequenza di byte come delimitatore, utilizzare il parametro **-sqdb** del comando **fteCreateTransfer** . Per ulteriori informazioni, consultare [parametro -sqdb](#).

Per impostazione predefinita, i messaggi creati da un trasferimento da file a messaggio sono persistenti. I messaggi possono essere impostati per essere non persistenti o per avere il valore di persistenza definito dalla coda di destinazione.

Se si specifica che un file è suddiviso in più messaggi, tutti i messaggi creati dal file hanno lo stesso ID gruppo IBM MQ . Se non si specifica che un file è suddiviso in più messaggi, solo un messaggio viene creato dal file e questo messaggio non ha l'ID gruppo IBM MQ impostato.

Se si stanno trasferendo file a messaggi di grandi dimensioni o a molti messaggi di piccole dimensioni, potrebbe essere necessario modificare alcune proprietà IBM MQ o Managed File Transfer . Per informazioni, consultare [Guida per l'impostazione degli attributi MQ e delle proprietà MFT associate alla dimensione del messaggio](#).

**Nota:** Se la coda di destinazione è una coda con cluster o un alias per una coda con cluster, si riceve un messaggio di errore durante il trasferimento di un file in una coda se l'output della proprietà dell'agent `enableClusterQueueInputnon` è stato impostato su `true`. Per ulteriori informazioni, fare riferimento a [Operazioni da eseguire se la coda di destinazione è una coda cluster o un alias per una coda cluster](#)

## Configurazione di un agent per eseguire trasferimenti da file a messaggi

Per impostazione predefinita, gli agent non possono eseguire trasferimenti da file a messaggio o da messaggio a file. Per abilitare questa funzione, è necessario impostare la proprietà dell'agent `enableQueueInputOutput` su `true`. Per abilitare la scrittura nelle code con cluster IBM MQ , è necessario impostare anche la proprietà dell'agente `enableClusterQueueInputOutput` su `true`.

### Informazioni su questa attività

Se si tenta di eseguire un trasferimento file - to - message a un agent di destinazione che non ha la proprietà `enableQueueInputOutput` impostata su `true`, il trasferimento non riesce. Il messaggio del log di trasferimento pubblicato nel gestore code di coordinamento contiene il messaggio seguente:

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

Per consentire all'agent di scrivere e leggere dalle code, effettuare le seguenti operazioni:

### Procedura

1. Arrestare l'agente di destinazione utilizzando il comando **Agent fteStop** .
2. Modificare il file `agent.properties` per includere la riga `enableQueueInputOutput=true`.  
Il file `agent.properties` si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name`.
3. Opzionale: Modificare il file `agent.properties` per includere la riga `enableClusterQueueInputOutput=true`. Il file `agent.properties` si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name`.
4. Avviare l'agent di destinazione utilizzando il comando **fteStartAgent** .

### Esempio: trasferimento di un singolo file in un singolo messaggio

È possibile specificare una coda come destinazione di trasferimento file utilizzando il parametro **-dq** con il comando **fteCreateTransfer** . Il file origine deve essere inferiore alla lunghezza massima del messaggio impostata sulla coda di destinazione. La coda di destinazione non deve trovarsi sullo stesso gestore code a cui si connette l'agent di destinazione, ma questi due gestori code devono essere in grado di comunicare.

### Informazioni su questa attività

Il file di origine è denominato `/tmp/single_record.txt` e si trova sullo stesso sistema dell'agente origine, `AGENT_NEPTUNE`. L'agent di origine, `AGENT_NEPTUNE`, utilizza il gestore code `QM_NEPTUNE`. L'agent di destinazione è `AGENT_VENUS` e questo agent si connette al gestore code `QM_VENUS`. La

coda di destinazione, RICEVIING\_QUEUE, si trova sul gestore code QM\_MERCURY. QM\_MERCURY è nella stessa rete IBM MQ del gestore code QM\_VENUS a cui è possibile accedere.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

Se la coda di destinazione si trova su un gestore code diverso da quello utilizzato dall'agente di destinazione, è necessario specificare il valore del parametro **-dq** nel seguente formato *nome\_coda@nome\_gestore\_code*. Se non si specifica *@queue\_manager\_name* nel valore, l'agente di destinazione presuppone che la coda di destinazione si trovi sul gestore code dell'agente di destinazione. L'eccezione si verifica quando la proprietà dell'agent di output `enableClusterQueueInput` è stata impostata su `true`. In questo caso, l'agent di destinazione utilizzerà le procedure di risoluzione IBM MQ standard per determinare dove si trova la coda.

L'agent di origine, AGENT\_NEPTUNE, legge i dati dal file `/tmp/single_record.txt` e trasferisce questi dati all'agent di destinazione, AGENT\_VENUS. L'agent di destinazione, AGENT\_VENUS, invia i dati a un messaggio persistente sulla coda `RECEIVING_QUEUE@QM_MERCURY`. Il messaggio non dispone di un ID gruppo IBM MQ impostato.

## Esempio: suddivisione di un singolo file in più messaggi per lunghezza

È possibile suddividere un file in più messaggi IBM MQ utilizzando il parametro **-qs** del comando **fteCreateTransfer**. Il file è suddiviso in sezioni a lunghezza fissa, ognuna delle quali viene scritta in un messaggio singolo.

### Informazioni su questa attività

Il file di origine è denominato `/tmp/source.file` e ha una dimensione di 36 KB. Il file di origine si trova sullo stesso sistema dell'agent di origine AGENT\_NEPTUNE. L'agent di origine, AGENT\_NEPTUNE, si connette al gestore code QM\_NEPTUNE. L'agent di destinazione è AGENT\_MERCURY, che si connette al gestore code QM\_MERCURY. La coda di destinazione, RICEVIING\_QUEUE, si trova anche sul gestore code QM\_MERCURY. Il trasferimento suddivide il file di origine in sezioni di dimensione pari a 1 KB e scrive ognuna di queste sezioni in un messaggio su RICEVIING\_QUEUE.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

L'agent di origine, AGENT\_NEPTUNE, legge i dati dal file `/tmp/source.file` e li trasferisce all'agent di destinazione, AGENT\_MERCURY. L'agent di destinazione, AGENT\_MERCURY, scrive i dati in 36 messaggi persistenti da 1 KB sulla coda `RICEVIING_QUEUE@QM_MERCURY`. Questi messaggi hanno tutti lo stesso ID gruppo IBM MQ e l'ultimo messaggio nel gruppo ha l'indicatore IBM MQ `LAST_MSG_IN_GROUP` impostato.

## Esempio: suddivisione di un file di testo in più messaggi utilizzando un delimitatore di espressione regolare

Trasferire un singolo file di testo a più messaggi suddividendo il file ad ogni corrispondenza di una determinata espressione regolare Java. A tale scopo, utilizzare il parametro **-dqdt** del comando **fteCreateTransfer**.

## Informazioni su questa attività

Il file viene suddiviso in sezioni di lunghezza variabile, ognuna delle quali viene scritta in un messaggio individuale. Il file di testo viene suddiviso in ogni punto in cui il testo nel file corrisponde a una determinata espressione regolare. Il file di origine è denominato `/tmp/names.text` e ha il seguente contenuto:

```
Jenny Jones,John Smith,Jane Brown
```

L'espressione regolare che specifica dove suddividere il file è il carattere virgola (`,`).

Il file di origine si trova sullo stesso sistema dell'agente origine `AGENT_NEPTUNE`, che si connette al gestore code `QM_NEPTUNE`. La coda di destinazione, `RICEVIING_QUEUE`, si trova sul gestore code `QM_MERCURY`. `QM_MERCURY` è anche il gestore code utilizzato dall'agent di destinazione `AGENT_MERCURY`. Il trasferimento suddivide il file di origine in sezioni e scrive ognuna di queste sezioni in un messaggio su `RICEVERE` la coda.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

L'agent di origine, `AGENT_NEPTUNE`, legge i dati dal file `/tmp/names.text` e li trasferisce all'agent di destinazione, `AGENT_MERCURY`. L'agent di destinazione, `AGENT_MERCURY`, scrive i dati in tre messaggi persistenti sulla coda `RICEVI_QUEUE`. Questi messaggi hanno tutti lo stesso ID gruppo IBM MQ e l'ultimo messaggio nel gruppo ha l'indicatore IBM MQ `LAST_MSG_IN_GROUP` impostato.

I dati nei messaggi sono i seguenti.

- Primo messaggio:

```
Jenny Jones
```

- Secondo messaggio:

```
John Smith
```

- Terzo messaggio:

```
Jane Brown
```

## Esempio: suddivisione di un file di testo con un delimitatore di espressione regolare e inclusione di un delimitatore nei messaggi

Trasferire un singolo file di testo a più messaggi suddividendo il file ad ogni corrispondenza di una determinata espressione regolare Java e includere la corrispondenza dell'espressione regolare nei messaggi risultanti. A tale scopo, utilizzare i parametri `-dqdt` e `-qi` del comando `fteCreateTransfer`.

## Informazioni su questa attività

Trasferire un singolo file di testo a più messaggi su una coda. Il file viene suddiviso in sezioni di lunghezza variabile, ognuna delle quali viene scritta in un messaggio individuale. Il file di testo viene suddiviso in ogni punto in cui il testo nel file corrisponde a una determinata espressione regolare. Il file di origine è denominato `/tmp/customers.text` e ha il seguente contenuto:

```
Customer name: John Smith
```

```
Customer contact details: john@example.net
Customer number: 314
```

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

L'espressione regolare che specifica dove suddividere il file è `Customer\snumber:\s\d+`, che corrisponde al testo "Numero cliente: " seguito da un qualsiasi numero di cifre. Le espressioni regolari specificate nella riga comandi devono essere racchiuse tra doppi apici per evitare che la shell dei comandi valuti l'espressione regolare. L'espressione regolare viene valutata come espressione regolare Java . Per ulteriori informazioni, consultare [Espressioni regolari utilizzate da MFT](#).

Per impostazione predefinita, il numero di caratteri che un'espressione regolare può corrispondere è impostato su cinque. L'espressione regolare utilizzata in questo esempio corrisponde a stringhe più lunghe di cinque caratteri. Per abilitare le corrispondenze che sono più lunghe di cinque caratteri, modificare il file delle proprietà dell'agent in modo da includere la proprietà **maxDelimiterMatchLength**.

Per default, il testo che corrisponde all'espressione regolare non viene incluso nei messaggi. Per includere il testo che corrisponde all'espressione regolare nei messaggi, come in questo esempio, utilizzare il parametro **-qi** . Il file di origine si trova sullo stesso sistema dell'agent di origine AGENT\_NEPTUNE, che si connette al gestore code QM\_NEPTUNE. La coda di destinazione, RICEVIING\_QUEUE, si trova sul gestore code QM\_MERCURY. QM\_MERCURY è anche il gestore code utilizzato dall'agent di destinazione AGENT\_MERCURY. Il trasferimento divide il file di origine in sezioni e scrive ognuna di queste sezioni in un messaggio su RICEVIING\_QUEUE.

## Procedura

1. Arrestare l'agent di destinazione utilizzando il comando seguente:

```
fteStopAgent AGENT_MERCURY
```

2. Aggiungere la seguente riga al file delle proprietà agent per AGENT\_MERCURY:

```
maxDelimiterMatchLength=25
```

**Nota:** L'aumento del valore di **maxDelimiterMatchLength** può ridurre le prestazioni.

3. Avviare l'agente di destinazione utilizzando il seguente comando:

```
fteStartAgent AGENT_MERCURY
```

4. Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
-dq RECEIVING_QUEUE
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

L'agent di origine, AGENT\_NEPTUNE, legge i dati dal file `/tmp/customers.text` e li trasferisce all'agent di destinazione, AGENT\_MERCURY. L'agent di destinazione, AGENT\_MERCURY, scrive i dati in tre messaggi persistenti sulla coda RICEVIING\_QUEUE. Questi messaggi hanno tutti lo stesso ID gruppo IBM MQ e l'ultimo messaggio nel gruppo ha l'indicatore IBM MQ LAST\_MSG\_IN\_GROUP impostato.

I dati nei messaggi sono i seguenti.

- Primo messaggio:

```
Customer name: John Smith
```

```
Customer contact details: john@example.net
Customer number: 314
```

- Secondo messaggio:

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

- Terzo messaggio:

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

## Esempio: impostazione delle proprietà del messaggio IBM MQ su un trasferimento file - a - messaggio

È possibile utilizzare il parametro **-qmp** nel comando **fteCreateTransfer** per specificare se le proprietà del messaggio IBM MQ sono impostate sul primo messaggio scritto nella coda di destinazione dal trasferimento. Le proprietà dei messaggi IBM MQ consentono a un'applicazione di selezionare i messaggi da elaborare o di recuperare informazioni su un messaggio senza accedere alle intestazioni IBM MQ MQMD (Message Descriptor) o MQRFH2 .

### Informazioni su questa attività

Includere il parametro **-qmp true** nel comando **fteCreateTransfer** . In questo esempio, l'ID utente MQMD dell'utente che inoltra il comando è `larmer`.

### Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true
-t text /tmp/source_file.txt
```

Le proprietà del messaggio IBM MQ del primo messaggio scritto dall'agent di destinazione, `AGENT_SATURN`, nella coda, `MY_QUEUE`, sul gestore code, `MyQM`, sono impostate sui seguenti valori:

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=text
usr.WMQFTESourceAgent=AGENT_JUPITER
usr.WMQFTEDestinationAgent=AGENT_SATURN
usr.WMQFTEFileName=source_file.txt
usr.WMQFTEFileSize=1024
usr.WMQFTEFileLastModified=1273740879040
usr.WMQFTEFileIndex=0
usr.WMQFTEMQmdUser=larmer
```

## Esempio: impostazione di proprietà definite dall'utente su un trasferimento file - a - messaggio

I metadati definiti dall'utente sono impostati come una proprietà del messaggio IBM MQ sul primo messaggio scritto nella coda di destinazione dal trasferimento. Le proprietà dei messaggi IBM MQ consentono a un'applicazione di selezionare i messaggi da elaborare o richiamare le informazioni su un messaggio senza accedere alle intestazioni IBM MQ MQMD (Message Descriptor) o MQRFH2 .

## Informazioni su questa attività

Includere i parametri `-qmp true` e `-md account=123456` nel comando `fteCreateTransfer`, per impostare la proprietà `usr.account` su `123456` nell'intestazione RFH2.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM
-qmp true -md account=123456 /tmp/source_file.txt
```

Oltre alla serie standard di proprietà del messaggio IBM MQ, la proprietà definita dall'utente è impostata nell'intestazione del messaggio del primo messaggio scritto dall'agente di destinazione, `AGENT_SATURN`, nella coda, `MY_QUEUE`, sul gestore code, `MyQM`. L'intestazione è impostata sul seguente valore:

```
usr.account=123456
```

Il prefisso `usr` viene aggiunto all'inizio del nome dei metadati definiti dall'utente.

### ***Esempio: aggiunta di una proprietà del messaggio definita dall'utente per il trasferimento da file a messaggio***

Se si sta utilizzando Managed File Transfer per i trasferimenti gestiti da messaggio a file, è possibile includere una proprietà del messaggio definita dall'utente per il messaggio risultante.

## Informazioni su questa attività

È possibile utilizzare uno dei seguenti metodi per definire una proprietà del messaggio personalizzata:

- Specificare il parametro `-md` nella richiesta di trasferimento. Per ulteriori informazioni, fare riferimento a [“Esempio: impostazione di proprietà definite dall'utente su un trasferimento file - a - messaggio” a pagina 255](#).
- Utilizzare un'attività Ant; è possibile utilizzare `fte:filecopy` o `fte:filemove`. Il seguente esempio è un'attività `fte:filecopy`:

```
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
  <property name="src.file" value="/home/user/file1.bin"/>
  <property name="dst.queue" value="TEST.QUEUE@qm2"/>
  <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="${FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="${src.file}" dstqueue="${dst.queue}"
dstmsgprops="true"/>

</fte:filecopy>

</target>
</project>
```

- Utilizzare un controllo risorse e la sostituzione delle variabili. Il seguente esempio mostra alcune attività di trasferimento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allOf>
        <condition>
          <fileMatch>
            <pattern>*.txt</pattern>
          </fileMatch>
        </condition>
      </allOf>
    </conditions>
  </triggerMatch>
  <tasks>
    <task>
      <name/>
      <transfer>
        <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
          <managedTransfer>
            <originator>
              <hostName>mqjason.raleigh.ibm.com.</hostName>
              <userID>administrator</userID>
            </originator>
            <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
            <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
            <transferSet priority="0">
              <metaDataSet>
                <metaData key="FileName">${FileName}</metaData>
              </metaDataSet>
              <item checksumMethod="MD5" mode="text">
                <source disposition="delete" recursive="false">
                  <file>${FilePath}</file>
                </source>
                <destination type="queue">
                  <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
                </destination>
              </item>
            </transferSet>
          </job>
        </managedTransfer>
      </request>
    </transfer>
  </task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>
```

### Attività correlate

[“Esempio: impostazione delle proprietà del messaggio IBM MQ su un trasferimento file - a - messaggio” a pagina 255](#)

È possibile utilizzare il parametro **-qmp** nel comando **fteCreateTransfer** per specificare se le proprietà del messaggio IBM MQ sono impostate sul primo messaggio scritto nella coda di destinazione dal trasferimento. Le proprietà dei messaggi IBM MQ consentono a un'applicazione di selezionare i messaggi da elaborare o di recuperare informazioni su un messaggio senza accedere alle intestazioni IBM MQ MQMD (Message Descriptor) o MQRFH2 .

## Riferimenti correlati

[attività fte: filecopy Ant](#)

[fte: filemove Ant attività](#)

## Errore di trasferimento da file a messaggio

Se un trasferimento file - a - messaggio ha esito negativo dopo che l'agent ha avviato la scrittura dei dati file nella coda di destinazione, l'agent scrive un messaggio nella coda per indicare a un'applicazione che utilizza i messaggi che si è verificato un errore.

Il messaggio scritto nella coda di destinazione se si verifica un errore:

- È vuoto
- Ha lo stesso ID gruppo IBM MQ del messaggio precedente scritto nella coda di destinazione dall'agente
- Ha impostato l'indicatore IBM MQ LAST\_MSG\_IN\_GROUP
- Contiene ulteriori proprietà del messaggio IBM MQ , se le proprietà del messaggio sono abilitate. Per ulteriori informazioni, consultare l'argomento [Proprietà dei messaggiMQ impostate da MFT sui messaggi scritti nelle code di destinazione.](#)

## Esempio

Un trasferimento viene richiesto immettendo il seguente comando:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE  
-qmp true -qs 1K /tmp/source1.txt
```

Il file `source1.txt` è 48 KB. Il trasferimento divide questo file in messaggi da 1 KB e scrive tali messaggi nella coda di destinazione `RICEVIING_QUEUE`.

Mentre il trasferimento è in corso, dopo che l'agente ha scritto 16 messaggi in `RICEVIING_QUEUE`, si verifica un errore sull'agente di origine.

L'agent scrive un messaggio vuoto in `RICEVIING_QUEUE`. Oltre alla serie standard di proprietà del messaggio, il messaggio vuoto ha la seguente serie di proprietà del messaggio:

```
usr.WMQFTEResultCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

**V 9.2.2** **V 9.2.0.2** Da IBM MQ 9.2.0 Fix Pack 2 e IBM MQ 9.2.2, quando un trasferimento da un file non riesce, a causa di un errore di controllo della dimensione del delimitatore, viene inviato un solo messaggio vuoto. Inoltre, le proprietà utente vengono aggiunte a questo messaggio se l'errore di trasferimento è dovuto al delimitatore che supera la dimensione impostata sull'agent di destinazione.

## Trasferimento dei dati dai messaggi ai file

La funzione `message - to - file` di Managed File Transfer consente di trasferire i dati da uno o più messaggi su una coda IBM MQ a un file, a un dataset (su z/OS) o a uno spazio file utente. Se si dispone di un'applicazione che crea o elabora messaggi IBM MQ , è possibile utilizzare la funzionalità `message - a - file` di Managed File Transfer per trasferire questi messaggi a un file su qualsiasi sistema nella rete Managed File Transfer .

Per informazioni sui trasferimenti da file a messaggi, consultare [“Trasferimento di dati dai file ai messaggi”](#) a pagina 250.



**Attenzione:** L'agent di origine per un trasferimento da messaggio a file non può essere un agent bridge di protocollo o un agent bridge Connect:Direct .

È possibile trasferire i dati del messaggio IBM MQ in un file. Sono supportati i seguenti tipi di trasferimento da messaggio a file:

- Da un singolo messaggio a un singolo file

- Da più messaggi a un singolo file
- Da più messaggi con lo stesso ID gruppo IBM MQ a un singolo file.
- Da più messaggi a un singolo file, incluso un delimitatore di testo o binario tra i dati di ciascun messaggio scritto nel file.

Se si stanno trasferendo file da messaggi di grandi dimensioni o da molti messaggi di piccole dimensioni, potrebbe essere necessario modificare alcune proprietà IBM MQ o Managed File Transfer . Per ulteriori informazioni, consultare [Guida per l'impostazione degli attributi di MQ e delle proprietà MFT associate alla dimensione del messaggio](#).

Da IBM MQ 9.1.0, in un trasferimento da messaggio a file, l'agent di origine esamina i messaggi dalla coda di origine, a differenza del GET distruttivo nelle versioni precedenti di IBM MQ. I messaggi vengono rimossi dalla coda di origine dopo che tutti i messaggi (in un gruppo se viene utilizzato il gruppo di messaggi) sono stati esaminati e i dati sono stati scritti nel file di destinazione. Ciò consente ai messaggi di rimanere nella coda di origine se un trasferimento non riesce o viene annullato. A causa di questa modifica, è necessario fornire anche l'autorizzazione a BROWSE insieme all'autorizzazione GET per eseguire i trasferimenti messaggio su file.

Da IBM MQ 9.0.0 Fix Pack 2 e IBM MQ 9.0.4, Managed File Transfer viene aggiornato per ripristinare il controllo di confronto, precedentemente rimosso da APAR IT18213, dell'ID trasferimento ed il valore dell'attributo groupId all'interno del payload XML della richiesta di trasferimento. Se questi due identificativi sono equivalenti, l'agent di origine utilizza l'identificatore come opzione di corrispondenza dell'identificativo del messaggio (in contrapposizione a un'opzione di corrispondenza dell'identificatore del gruppo) per il primo tentativo MQGET effettuato sulla coda di input per il trasferimento da messaggio a file.

## Configurazione di un agent per eseguire trasferimenti da messaggio a file

Per impostazione predefinita, gli agent non possono eseguire trasferimenti da messaggio a file o da file a messaggio. Per abilitare questa funzione è necessario impostare la proprietà dell'agente enableQueueInputOutput su true.

### Informazioni su questa attività

Se si tenta di eseguire un messaggio per il trasferimento file da un agent di origine che non ha la proprietà enableQueueInputOutput impostata su true, il trasferimento non riesce. Il messaggio del log di trasferimento pubblicato nel gestore code di coordinamento contiene il messaggio seguente:

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.
The agent must have enableQueueInputOutput=true set in the agent.properties file
to support transferring from a queuee.
```

Per consentire all'agent di scrivere e leggere dalle code, effettuare le seguenti operazioni:

### Procedura

1. Arrestare l'agente di origine utilizzando il comando **fteStopAgent** .
2. Modificare il file agent .properties per includere la riga enableQueueInputOutput=true.  
Il file agent .properties si trova nella directory *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/source\_agent\_name*.
3. Avviare l'agente di origine utilizzando il comando **fteStartAgent** .

### Esempio: trasferimento da una coda a un file singolo

È possibile specificare una coda di IBM MQ come origine di trasferimento file utilizzando il parametro **-sq** con il comando **fteCreateTransfer** .

## Informazioni su questa attività

I dati di origine sono contenuti in tre messaggi sulla coda *START\_QUEUE*. Questa coda deve essere sul gestore code dell'agente di origine, *QM\_NEPTUNE*.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
                  -da AGENT_VENUS -df /out/three_to_one.txt
                  -sq START_QUEUE
```

I dati nei messaggi sulla coda *START\_QUEUE* vengono scritti nel file */out/three\_to\_one.txt* sul sistema su cui è in esecuzione *AGENT\_VENUS*.

## Esempio: trasferimento di un gruppo di messaggi da una coda a un file singolo

È possibile specificare un singolo gruppo completo su una coda IBM MQ come origine di un trasferimento file utilizzando i parametri **-sq** e **-sqgi** con il comando **fteCreateTransfer**.

## Informazioni su questa attività

In questo esempio, ci sono dieci messaggi sulla coda *START\_QUEUE*. Questa coda deve essere sul gestore code dell'agente di origine, *QM\_NEPTUNE*. I primi tre messaggi appartengono a un gruppo con IBM MQ ID gruppo 41424b3ef3a22020202020202020202020202020202020201111; questo gruppo non è un gruppo completo. I successivi cinque messaggi appartengono ad un gruppo con IBM MQ ID gruppo 41424b3ef3a22020202020202020202020202020202020202222; questo gruppo è completo. I restanti due messaggi appartengono ad un gruppo con IBM MQ ID gruppo 41424b3ef3a22020202020202020202020202020202020203333; questo gruppo è completo.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS
                  -df /out/group.txt -sqgi -sq START_QUEUE
```

I dati nei messaggi che appartengono al primo gruppo completo sulla coda *START\_QUEUE*, il gruppo con IBM MQ ID gruppo 41424b3ef3a22020202020202020202020202020202020202222, vengono scritti nel file */out/group.txt* sul sistema su cui è in esecuzione *AGENT\_VENUS*.

## Esempio: inserimento di un delimitatore di testo prima dei dati da ciascun messaggio

Quando si esegue il trasferimento in modalità testo da una coda di origine a un file, è possibile specificare che un delimitatore di testo viene inserito prima dei dati dei singoli messaggi utilizzando i parametri **-sq**, **-sqdt** e **-sqdp** con il comando **fteCreateTransfer**.

## Informazioni su questa attività

In questo esempio, ci sono quattro messaggi sulla coda *START\_QUEUE*. Questa coda è sul gestore code dell'agente di origine, *QM\_NEPTUNE*. Il delimitatore di testo da inserire prima dei dati da ciascun messaggio può essere espresso come una stringa letterale Java, ad esempio `\n\u002D\u002D\u002D\n`.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt  
-t text -sqdt "\n\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

Il delimitatore di testo viene aggiunto all'inizio dei dati da ognuno dei quattro messaggi su START\_QUEUE dall'agent di origine, AGENT\_NEPTUNE. Questi dati vengono scritti sul file di destinazione, /out/output.txt.

## Esempio: inserimento di un delimitatore binario dopo i dati da ciascun messaggio

Quando si esegue il trasferimento in modalità binaria da una coda di origine a un file, è possibile specificare che venga inserito un delimitatore binario dopo i dati dei singoli messaggi utilizzando i parametri **-sq**, **-sqdbe** **-sqdp** con il comando **fteCreateTransfer**.

### Informazioni su questa attività

In questo esempio, ci sono tre messaggi sulla coda START\_QUEUE. Questa coda è sul gestore code dell'agente di origine, *QM\_NEPTUNE*. Il delimitatore binario da inserire dopo i dati da ciascun messaggio deve essere espresso come un elenco separato da virgole di byte esadecimali, ad esempio *x34, xE7, xAE*.

## Procedura

Immettere il seguente comando:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

Il delimitatore binario viene aggiunto ai dati da ognuno dei tre messaggi su START\_QUEUE dall'agent di origine AGENT\_NEPTUNE. Questi dati vengono scritti sul file di destinazione, /out/binary.file.

## Monitoraggio di una coda e utilizzo della sostituzione di variabili

È possibile monitorare una coda e trasferire messaggi dalla coda monitorata a un file utilizzando il comando **fteCreateMonitor**. Il valore di qualsiasi proprietà del messaggio IBM MQ nel primo messaggio da leggere dalla coda monitorata può essere sostituito nella definizione XML dell'attività e utilizzato per definire il funzionamento del trasferimento.

### Informazioni su questa attività

In questo esempio, l'agent di origine è denominato AGENT\_VENUS, che si connette a QM\_VENUS. La coda monitorata da AGENT\_VENUS è denominata START\_QUEUE e si trova su QM\_VENUS. L'agent esegue il polling della coda ogni 30 minuti.

Quando un gruppo completo di messaggi viene scritto nella coda, l'attività di monitoraggio invia il gruppo di messaggi a uno dei diversi agent di destinazione, tutti connessi al gestore code QM\_MARS. Il nome del file a cui viene trasferito il gruppo di messaggi viene definito dalla proprietà IBM MQ message *usr.fileName* sul primo messaggio del gruppo. Il nome dell'agent a cui viene inviato il gruppo di messaggi è definito dalla IBM MQ proprietà del messaggio *usr.toAgent* sul primo messaggio nel gruppo. Se l'intestazione *usr.toAgent* non è impostata, il valore predefinito da utilizzare per l'agent di destinazione è AGENT\_MAGENTA.

Quando si specificano *useGroups="true"*, se non si specifica anche *groupId="\$GROUPID"*, il trasferimento acquisisce solo il primo messaggio sulla coda. Ad esempio, se si sta utilizzando la

sostituzione della variabile per generare il fileName, è possibile che il contenuto di a .txt non sia corretto. Ciò è dovuto al fatto che fileName viene generato dal monitoraggio, ma il trasferimento in realtà riceve un messaggio che non è quello che dovrebbe generare il file denominato fileName.

## Procedura

1. Creare l'attività XML che definisce l'attività che il monitoraggio esegue quando viene attivato.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Le variabili sostituite con i valori delle intestazioni dei messaggi IBM MQ sono evidenziati in **grassetto**. Questa attività XML viene salvata nel file /home/USER1/task.xml

2. Creare un monitoraggio risorse per monitorare la coda START\_QUEUE.  
Inoltrare il seguente comando:

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Un utente o un programma scrive un gruppo di messaggi nella coda START\_QUEUE.  
Il primo messaggio in questo gruppo ha le seguenti proprietà del messaggio IBM MQ :

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. Il controllo viene attivato quando viene scritto il gruppo completo. L'agent sostituisce le proprietà del messaggio IBM MQ nell'XML dell'attività.  
Ciò determina la trasformazione dell'attività XML in:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
```

```
<file>/reports/larmer.rpt</file>
  </destination>
</item>
</transferSet>
</managedTransfer>
</request>
```

## Risultati

Viene effettuato il trasferimento definito dall'attività XML. Il gruppo completo di messaggi letti da START\_QUEUE da AGENT\_VENUS viene scritto in un file denominato /reports/larmer.rpt sul sistema su cui è in esecuzione AGENT\_VIOLET.

## Operazioni successive

### Trasferimento di ogni messaggio in un file separato

Se si desidera monitorare una coda e trasferire ogni messaggio in un file separato, è possibile utilizzare una tecnica simile a quella descritta in precedenza in questo argomento.

1. Creare il monitoraggio come descritto in precedenza, specificando il parametro **-tr completeGroups** nel comando **fteCreateMonitor**.
2. Nell'XML dell'attività specificare quanto segue:

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

Tuttavia, quando si inserono i messaggi nella coda di origine, non inserirli in un gruppo IBM MQ. Aggiungere le proprietà del messaggio IBM MQ a ciascun messaggio. Ad esempio, specificare la proprietà `usr.filename` con un valore di nome file univoco per ciascun messaggio. Ciò fa sì che Managed File Transfer Agent consideri ciascun messaggio sulla coda di origine come un gruppo separato.

## Esempio: errore di trasferimento da messaggio a file utilizzando le proprietà del messaggio IBM MQ

È possibile causare un errore nel trasferimento file di un messaggio impostando la proprietà del messaggio `usr.UserReturnCode` IBM MQ su un valore diverso da zero. È inoltre possibile specificare ulteriori informazioni sul motivo dell'errore impostando la proprietà del messaggio `usr.UserSupplement` IBM MQ.

## Informazioni su questa attività

In questo esempio, è in corso un trasferimento tra la coda INPUT\_QUEUE e il file /home/user/output.file.

Un utente sta creando messaggi e li sta collocando nella coda INPUT\_QUEUE. L'agent di origine sta utilizzando i messaggi dalla coda INPUT\_QUEUE e sta inviando i dati di trasferimento all'agent di destinazione. L'agent di destinazione sta scrivendo questi dati nel file /home/user/output.file.

L'utente che scrive i messaggi nella coda INPUT\_QUEUE desidera arrestare il trasferimento in corso ed eliminare i dati già scritti nel file di destinazione.

## Procedura

1. L'utente scrive un messaggio nella coda INPUT\_QUEUE con le seguenti proprietà del messaggio IBM MQ impostate:

```
usr.UserReturnCode=1
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. L'agent di origine legge le proprietà del messaggio IBM MQ e arresta l'elaborazione dei messaggi dalla coda. L'agent di destinazione elimina tutti i dati file che sono stati scritti nella directory di destinazione.

3. L'agent di origine invia un messaggio di log di trasferimento al gestore code di coordinamento che riporta l'errore di trasferimento.

Il messaggio contiene le seguenti informazioni:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020207e970d4920008702" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T21:28:09.593Z">progress</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </destinationAgent>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <transferSet index="0" size="1"
    startTime="2008-11-02T21:28:09.281Z"
    total="1">
    <item mode="binary">
      <source>
        <queue>INPUT_QUEUE@QM1</queue>
      </source>
      <destination exist="error">
        <file>/home/user/output.file</file>
      </destination>
      <status resultCode="1">
        <supplement>Cancelling transfer - sent wrong data.</supplement>
      </status>
    </item>
  </transferSet>
</transaction>
```

## Il bridge di protocollo

Il bridge di protocollo consente alla rete Managed File Transfer (MFT) di accedere ai file memorizzati su un server di file esterno alla rete MFT, nel dominio locale o in un'ubicazione remota. Questo server di file può utilizzare i protocolli di rete FTP, FTPS o SFTP. Ogni server di file richiede almeno un agent dedicato. L'agent dedicato è noto come agent bridge di protocollo. Un agent bridge può interagire con più server di file.

Il bridge di protocollo è disponibile come parte del componente Service di Managed File Transfer. È possibile disporre di più agent dedicati su un singolo sistema su cui è in esecuzione MFT che si collegano a server di file diversi.

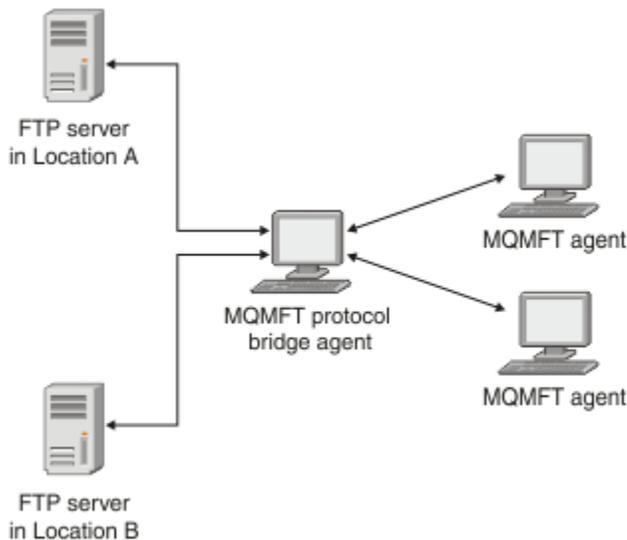
È possibile utilizzare un agent bridge di protocollo per trasferire file a più endpoint contemporaneamente. MFT fornisce un file denominato `ProtocolBridgeProperties.xml` che è possibile modificare per definire i diversi server di file del protocollo a cui si desidera trasferire i file. Il comando **fteCreateBridgeAgent** aggiunge automaticamente i dettagli del server di file del protocollo predefinito a `ProtocolBridgeProperties.xml`. Questo file è descritto in [Formato file delle proprietà del bridge di protocollo](#).

È possibile utilizzare l'agent bridge di protocollo per eseguire queste operazioni:

- Caricare i file dalla rete MFT a un server remoto utilizzando FTP, FTPS o SFTP.
- Scaricare i file da un server remoto, utilizzando FTP, FTPS o SFTP, nella rete MFT

**Nota:** L'agent bridge di protocollo può supportare solo server FTP, FTPS o SFTP che consentono l'accesso ai file tramite il percorso file assoluto. Se in una richiesta di trasferimento viene specificato un percorso file relativo, l'agente bridge di protocollo tenterà di convertire il relativo percorso in un percorso file assoluto basato sulla directory home utilizzata per accedere al server del protocollo. I server di protocollo

che consentono l'accesso ai file basati solo sulla directory corrente non sono supportati dall'agente bridge di protocollo.



Il diagramma mostra due server FTP, in diverse posizioni. I server FTP vengono utilizzati per scambiare file con gli agenti Managed File Transfer. L'agente bridge di protocollo si trova tra i server FTP e il resto della rete MFT ed è configurato per comunicare con entrambi i server FTP.

Accertarsi di disporre di un altro agente nella rete MFT in aggiunta all'agente bridge di protocollo. L'agente bridge di protocollo è un bridge solo per il server FTP, FTPS o SFTP e non scrive i file trasferiti sul disco locale. Se si desidera trasferire file da o verso il server FTP, FTPS o SFTP, è necessario utilizzare l'agente bridge di protocollo come destinazione o origine per il trasferimento file (che rappresenta il server FTP, FTPS o SFTP) e un altro agente standard come origine o destinazione corrispondente.

Quando si trasferiscono i file utilizzando il bridge di protocollo, il bridge deve avere l'autorizzazione a leggere la directory di origine o di destinazione contenente i file che si desidera trasferire. Ad esempio, se si desidera trasferire i file dalla directory `/home/fte/bridge` che dispone solo delle autorizzazioni di esecuzione (`d -- x -- x -- x`), i trasferimenti che si tentano da questa directory non riescono con il seguente messaggio di errore:

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

## Configurazione di un agente bridge di protocollo

Un agente bridge di protocollo è come un agente MFT standard. Creare un agente bridge di protocollo utilizzando il comando `fteCreateBridgeAgent`. È possibile configurare un agente bridge di protocollo utilizzando il file `ProtocolBridgeProperties.xml`, descritto in [Formato file delle proprietà bridge di protocollo](#). Se si utilizza una versione precedente, configurare l'agente utilizzando le proprietà bridge di protocollo specifiche descritte in [Proprietà agent avanzate: bridge di protocollo](#) e [Proprietà agent avanzate: registrazione agent bridge di protocollo](#). Per tutte le versioni, è anche possibile configurare un'associazione credenziali come descritto in ["Associazione delle credenziali per un server di file" a pagina 273](#). Dopo aver configurato un agente bridge di protocollo per un particolare server di file di protocollo, è possibile utilizzare tale agente solo per tale scopo.

## Ripristino bridge di protocollo

Se l'agente bridge di protocollo non è in grado di connettersi al server di file perché il server di file non è disponibile, tutte le richieste di trasferimento file vengono accodate fino a quando il server di file non diventa disponibile. Se l'agente bridge di protocolli non è in grado di connettersi al server di file perché l'agente utilizza le credenziali errate, il trasferimento ha esito negativo e il messaggio del log di trasferimento riflette questo errore. Se l'agente bridge di protocollo viene terminato per qualsiasi ragione,

tutti i trasferimenti file richiesti vengono conservati e continuano quando il bridge di protocollo viene riavviato.

Durante il trasferimento file, i file vengono generalmente scritti come file temporanei nella destinazione e vengono ridenominati quando il trasferimento è completo. Tuttavia, se la destinazione di trasferimento è un server di file di protocollo configurato come scrittura limitata (gli utenti possono caricare i file sul server di file di protocollo ma non possono modificare in alcun modo i file caricati; in effetti gli utenti possono scrivere una sola volta), i file trasferiti vengono scritti direttamente nella destinazione. Ciò significa che se si verifica un problema durante il trasferimento, i file scritti parzialmente rimangono sul server di file del protocollo di destinazione e Managed File Transfer non può eliminare o modificare tali file. In questa situazione, il trasferimento non riesce.

## Definizione delle proprietà per i server di file del protocollo utilizzando il file ProtocolBridgeProperties.xml

Definire le proprietà di uno o più server di file di protocollo a cui si desidera trasferire i file utilizzando il file ProtocolBridgeProperties.xml, fornito da Managed File Transfer nella directory di configurazione dell'agent.

### Informazioni su questa attività

Il comando **fteCreateBridgeAgent** crea il file ProtocolBridgeProperties.xml nella directory di configurazione dell'agente *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/bridge\_agent\_name*. Il comando crea anche una voce nel file per il server del file del protocollo predefinito, se è stato specificato un valore predefinito quando è stato eseguito il comando.

Il messaggio BFGCL0392I fornisce l'ubicazione del file ProtocolBridgeProperties.xml.

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.

Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or
<tns:sftpServer>
element - depending on the protocol used to communicate with the server. When the protocol
bridge agent participates in a managed file transfer it will determine which server to used
based on
the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/
file.txt' would
be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name
is compared
to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element
in this
XML document and the first match is used to determine which protocol server the protocol bridge
agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this
XML
document can specify a default server to use for the managed file transfer. To specify a
default server use the <tns:defaultServer> element as the first element inside the
<tns:serverProperties>
element. The default server will be used whenever the protocol bridge agent participates in
a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element
contains
attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each serverProperties
definition. This
element contains a path to a file containing credentials to be used when connecting to defined
servers.
```

An example ProtocolBridgeProperties.xml file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
```

```

<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
<tns:defaultServer name="myFTPserver" />
  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234"
platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  listFormat="unix" limitedWrite="false">
    <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"
      maxReconnectRetry="2" reconnectWaitPeriod="10"
      maxSessions="60" socketTimeout="30" />
  </tns:ftpServer>
  <tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false" ftpsType="explicit"
  trustStore="C:\FTE\keystores\myFTPSServer\FTPSKeyStore.jks"
  trustStorePassword="password">
    <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>
  </tns:ftpsServer>
  <tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  limitedWrite="false">
    <tns:limits connectionTimeout="60"/>
  </tns:sftpServer>
</tns:serverProperties>

```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to the command line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```

-->
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
ProtocolBridgeProperties.xsd">

```

<!-- By default the location of the credentials file is in the home directory of the user that started the -->

<!-- protocol bridge agent. If you wish to specify a different location use the credentialsFile element to -->

<!-- do this. For

example:

```

<!-- <tns:credentialsFile path="/test/
ProtocolBridgeCredentials.xml"/>

```

-->

-->

```

    <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM" />
    <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
platform="unix"
        timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
        listFormat="unix" limitedWrite="false" />

    <!-- Define servers here -->
</tns:serverProperties>

```

Il comando può produrre il seguente messaggio: BFGCL0532I:

Perché l'agent funzioni, è necessario creare manualmente un file delle credenziali aggiuntivo. Per impostazione predefinita, questo file è denominato ProtocolBridgeCredentials.xml e si trova nella home della directory dell'utente che avvia l'agente. Ad esempio, se questo utente ha avviato l'agent l'ubicazione è: \$HOME/ProtocolBridgeCredentials.xml

Se si utilizza un file delle credenziali:

1. Consultare il seguente testo per ulteriori informazioni su come crearne uno.
2. Il file delle credenziali deve trovarsi in una directory con autorizzazioni limitate. Ad esempio, non ci deve essere alcun accesso in lettura per altri utenti.
3. Specificare l'ubicazione della directory per il file delle credenziali nella variabile di ambiente \$HOME per l>ID utente dell'agent avviato oppure modificare il file ProtocolBridgeProperties.xml e specificare l'ubicazione in:

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

Se si desidera aggiungere ulteriori server di protocollo non predefiniti, modificare questo file per definirne le proprietà. Questo esempio aggiunge un server FTP aggiuntivo.

**Nota:** L'agent bridge di protocollo non supporta il blocco file. Ciò è dovuto al fatto che Managed File Transfer non supporta il meccanismo di blocco dei file su un file server.

## Procedura

1. Definire un server di file di protocollo inserendo le righe seguenti nel file come elemento child di <tns:serverProperties>:

```

<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
platform="windows"
        timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
        listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>

```

2. Modificare quindi il valore degli attributi:

- name è il nome del server di file del protocollo
- host è il nome host o l'indirizzo IP del server di file del protocollo
- port è il numero di porta del server di file del protocollo
- platform è la piattaforma su cui viene eseguito il server del file di protocollo
- timeZone è il fuso orario in cui viene eseguito il server del file di protocollo
- locale è la lingua utilizzata sul server di file del protocollo
- fileEncoding è la codifica dei caratteri del server di file del protocollo
- listFormat è il formato di elenco file restituito dal server di file del protocollo
- limitedWrite determina se seguire la modalità predefinita durante la scrittura su un server di file, che consiste nel creare un file temporaneo e ridenominare tale file una volta completato il trasferimento. Per un server di file configurato come sola scrittura, il file viene creato direttamente con il nome finale. Il valore di questa proprietà può essere true o false. L'attributo limitedWrite e la proprietà dell'agent doNotUseTempOutputFile vengono utilizzati insieme nel caso di agent bridge di protocollo. Se si desidera utilizzare i file temporanei, non è necessario impostare il valore

di `doNotUseTempOutputFile` è necessario impostare il valore di `limitedWrite` su `false`. Qualsiasi altra combinazione di impostazioni significa che i file temporanei non verranno utilizzati.

- `maxListFileNames` è il numero massimo di nomi raccolti durante la scansione di una directory sul server di file del protocollo per i nomi file.
- `maxListDirectoryLevels` è il numero massimo di livelli di directory da ricorre quando si esegue la scansione di una directory sul server di file del protocollo per i nomi file.

Per ulteriori dettagli su questi attributi, inclusi se sono obbligatori o facoltativi e i loro valori predefiniti, consultare [Formato file delle proprietà del bridge di protocollo](#).

### Riferimenti correlati

[Formato file delle proprietà bridge di protocollo](#)

[Espressioni regolari utilizzate da MFT](#)

## Ricerca delle proprietà del server di file del protocollo: ProtocolBridgePropertiesExit2

Se si dispone di un numero elevato di server di file di protocollo, è possibile implementare l'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` per ricercare le proprietà del server di file di protocollo a cui si fa riferimento nei trasferimenti. È possibile implementare questa interfaccia preferendo mantenere un file `ProtocolBridgeProperties.xml`.

### Informazioni su questa attività

Managed File Transfer fornisce un'uscita utente di esempio che ricerca le proprietà del server di file di protocollo. Per ulteriori informazioni, consultare ["Utilizzo dell'uscita utente di esempio per la ricerca delle proprietà del server di file del protocollo"](#) a pagina 270.

Qualsiasi uscita utente che ricerca le proprietà del bridge di protocollo deve implementare l'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2`. Per ulteriori informazioni, consultare [ProtocolBridgePropertiesExit.java](#) *interface*.

È possibile concatenare più uscite delle proprietà del server di protocollo in modo simile ad altre uscite utente. Le uscite vengono richiamati nell'ordine in cui vengono specificate utilizzando la proprietà `protocolBridgePropertiesExitClasses` del file delle proprietà dell'agente. I metodi di inizializzazione vengono restituiti tutti separatamente e se uno o più restituisce un valore `false`, l'agent non viene avviato. L'errore viene riportato nella registrazione eventi dell'agent.

Viene restituito un unico risultato complessivo per i metodi `getProtocolServerProperties` di tutte le uscite. Se il metodo restituisce un oggetto proprietà come codice risultato, questo valore è il risultato restituito e i metodi `getProtocolServerProperties` delle uscite successive non vengono richiamati. Se il metodo restituisce un valore `null` come codice di risultato, viene richiamato il metodo `getProtocolServerProperties` dell'uscita successiva. Se non vi è alcuna uscita successiva, viene restituito il risultato `null`. Un codice di risultato globale `null` viene considerato come un errore di ricerca dall'agente bridge di protocollo.

Si consiglia di utilizzare l'interfaccia `ProtocolBridgePropertiesExit2.java`, ma per informazioni sull'interfaccia `ProtocolBridgePropertiesExit.java`, consultare ["Ricerca delle proprietà del server di file del protocollo: ProtocolBridgePropertiesExit"](#) a pagina 271.

Per eseguire l'uscita, completare la seguente procedura:

### Procedura

1. Compilare l'uscita utente delle proprietà del server di protocollo.
2. Creare un file JAR (Java archive) contenente l'exit compilata e la relativa struttura del package.
3. Inserire il file JAR contenente la classe di uscita nella directory `exits` dell'agent bridge di protocollo. Questa directory si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.

4. Modificare il file delle proprietà dell'agent bridge di protocollo per includere la proprietà `protocolBridgePropertiesExitClasses`. Per il valore di questa proprietà, specificare un elenco separato da virgole di classi che implementano un'uscita utente delle proprietà del server bridge di protocollo. Le classi di uscita vengono richiamati nell'ordine in cui vengono specificate in questo elenco. Per ulteriori informazioni, consultare [Il file MFT `agent.properties`](#).
5. Facoltativamente, è possibile specificare la proprietà `protocolBridgePropertiesConfiguration`. Il valore specificato per questa proprietà viene passato come stringa al metodo `initialize()` delle classi di uscita specificate da `protocolBridgePropertiesExitClasses`. Per ulteriori informazioni, consultare [Il file MFT `agent.properties`](#).

### **Utilizzo dell'uscita utente di esempio per la ricerca delle proprietà del server di file del protocollo**

Managed File Transfer fornisce un'uscita utente di esempio che ricerca le proprietà del server di file di protocollo.

### **Informazioni su questa attività**

Un'uscita utente di esempio che ricerca le proprietà del bridge di protocollo viene fornita nella directory `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` e nell'argomento [Uscita utente delle proprietà del bridge di protocollo](#).

L'uscita `SamplePropertiesExit2.java` legge un file di proprietà che contiene le proprietà per i server di protocolli. Il formato di ciascuna voce nel file delle proprietà è il seguente:

```
serverName=type://host:port
```

L'ubicazione del file delle proprietà viene presa dalla proprietà dell'agent bridge di protocollo `protocolBridgePropertiesConfiguration`.

Per eseguire l'uscita utente di esempio, completare la seguente procedura:

### **Procedura**

1. Compilare il file `SamplePropertiesExit2.java`.
2. Creare un file JAR contenente l'uscita compilata e la relativa struttura del package.
3. Inserire il file JAR nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits`.
4. Modificare il file `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` in modo che contenga la riga:

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. Creare un file delle proprietà bridge di protocollo, ad esempio `protocol_bridge_properties.properties`, nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent`. Modificare questo file per includere le voci nel formato:

```
serverName=type://host:port
```

6. Modificare il file `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` in modo che contenga la riga:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

È necessario utilizzare il percorso assoluto del file `protocol_bridge_properties.properties`.

7. Avviare l'agent bridge di protocollo utilizzando il comando **fteStartAgent** .

### Concetti correlati

[“Il bridge di protocollo” a pagina 264](#)

Il bridge di protocollo consente alla rete Managed File Transfer (MFT) di accedere ai file memorizzati su un server di file esterno alla rete MFT , nel dominio locale o in un'ubicazione remota. Questo server di file può utilizzare i protocolli di rete FTP, FTPS o SFTP. Ogni server di file richiede almeno un agent dedicato. L'agent dedicato è noto come agent bridge di protocollo. Un agent bridge può interagire con più server di file.

### Riferimenti correlati

[Interfaccia ProtocolBridgePropertiesExit.java](#)

[User exit delle proprietà del bridge di protocollo di esempio](#)

[Il file MFT agent.properties](#)

[fteCreateBridgeAgent \(creazione e configurazione di un agent bridge di protocollo MFT \)](#)

### **Ricerca delle proprietà del server di file del protocollo: ProtocolBridgePropertiesExit**

Se si dispone di un numero elevato di server di file di protocollo, è possibile implementare l'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` per ricercare le proprietà del server di file di protocollo a cui si fa riferimento nei trasferimenti.

### Informazioni su questa attività

È possibile implementare l'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` piuttosto che gestire un file `ProtocolBridgeProperties.xml` . Si consiglia di utilizzare l'interfaccia `ProtocolBridgePropertiesExit2.java` , ma è supportata anche `ProtocolBridgePropertiesExit.java`. Se si dispone di un'implementazione esistente dell'interfaccia `PropertiesExit.java` di `ProtocolBridgeda IBM WebSphere MQ File Transfer Edition`, è possibile utilizzarla in `IBM WebSphere MQ 7.5` o in una versione successiva. Il metodo di ubicazione `getCredentialin ProtocolBridgePropertiesExit2.java` utilizza l'ubicazione predefinita del file `ProtocolBridgeCredentials.xml` , che è la `directory home`.

**Nota:** `IBM WebSphere MQ File Transfer Edition (FTE)` non è più un prodotto supportato. Per eseguire la migrazione da FTE al componente MFT (Managed File Transfer) in `IBM MQ`, consultare [Migrazione di MFT \(Managed File Transfer\)](#).

Qualsiasi uscita utente che ricerca le proprietà bridge di protocollo deve implementare l'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit`:

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *        The values of properties defined for the protocol bridge.
     *        These values can only be read, they cannot be updated by the
     *        implementation.
     * @return {@code true} if the initialization is successful and {@code
     *         false} if unsuccessful. If {@code false} is returned from an exit
```

```

*         the protocol bridge agent will not start.
*/
public boolean initialize(final Map<String, String> bridgeProperties);

/**
 * Obtains a set of properties for the specified protocol server name.
 * <p>
 * The returned {@link Properties} must contain entries with key names
 * corresponding to the constants defined in
 * {@link ProtocolServerPropertyConstants} and in particular must include an
 * entry for all appropriate constants described as required.
 *
 * @param protocolServerName
 *         The name of the protocol server whose properties are to be
 *         returned. If a null or a blank value is specified, properties
 *         for the default protocol server are to be returned.
 * @return The {@link Properties} for the specified protocol server, or null
 *         if the server cannot be found.
 */
public Properties getProtocolServerProperties(
    final String protocolServerName);

/**
 * Invoked once when a protocol bridge agent is shut down. It is intended to
 * release any resources that were allocated by the exit.
 *
 * @param bridgeProperties
 *         The values of properties defined for the protocol bridge.
 *         These values can only be read, they cannot be updated by the
 *         implementation.
 */
public void shutdown(final Map<String, String> bridgeProperties);
}

```

È possibile concatenare più uscite delle proprietà del server di protocollo in modo simile ad altre uscite utente. Le uscite vengono richiamati nell'ordine in cui vengono specificate utilizzando la proprietà `protocolBridgePropertiesExitClasses` del file delle proprietà dell'agente. I metodi di inizializzazione vengono restituiti tutti separatamente e se uno o più restituisce un valore false, l'agent non viene avviato. L'errore viene riportato nella registrazione eventi dell'agente.

Viene restituito un unico risultato complessivo per i metodi `getProtocolServerProperties` di tutte le uscite. Se il metodo restituisce un oggetto proprietà come codice risultato, questo valore è il risultato restituito e i metodi `getProtocolServerProperties` delle uscite successive non vengono richiamati. Se il metodo restituisce un valore null come codice di risultato, viene richiamato il metodo `getProtocolServerProperties` dell'uscita successiva. Se non vi è alcuna uscita successiva, viene restituito il risultato null. Un codice di risultato globale null viene considerato come un errore di ricerca dall'agente bridge di protocollo.

## Procedura

Per eseguire l'uscita, completare la seguente procedura:

1. Compilare l'uscita utente delle proprietà del server di protocollo.
2. Creare un file JAR ( Java archive) contenente l'exit compilata e la relativa struttura del package.
3. Inserire il file JAR contenente la classe di uscita nella directory `exits` dell'agent bridge di protocollo.

Questa directory si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.

4. Modificare il file delle proprietà dell'agent bridge di protocollo per includere la proprietà `protocolBridgePropertiesExitClasses`.

Per il valore di questa proprietà, specificare un elenco separato da virgole di classi che implementano un'uscita utente delle proprietà del server bridge di protocollo. Le classi di uscita vengono richiamati nell'ordine in cui vengono specificate in questo elenco. Per ulteriori informazioni, consultare [Il file MFT agent.properties](#).

5. Facoltativamente, è possibile specificare la proprietà `protocolBridgePropertiesConfiguration`.

Il valore specificato per questa proprietà viene passato come stringa al metodo `initialize ()` delle classi di uscita specificate da `protocolBridgePropertiesExitClasses`. Per ulteriori informazioni, consultare [Il file MFT `agent.properties`](#).

## Associazione delle credenziali per un server di file

Associare le credenziali utente in Managed File Transfer alle credenziali utente sul server di file utilizzando la funzione di associazione credenziali predefinita dell'agent bridge di protocollo o scrivendo la propria uscita utente. Managed File Transfer fornisce un'uscita utente di esempio che esegue l'associazione delle credenziali utente.

### **Associazione delle credenziali per un server di file utilizzando il file `ProtocolBridgeCredentials.xml`**

Associare le credenziali utente in Managed File Transfer alle credenziali utente nel file server utilizzando la funzione di associazione credenziali predefinita dell'agent bridge di protocollo. Managed File Transfer fornisce un file XML che è possibile modificare per includere le informazioni sulle credenziali.

## Informazioni su questa attività

Il file `ProtocolBridgeCredentials.xml` deve essere creato manualmente dall'utente. Per impostazione predefinita, il percorso di questo file è la directory home dell'utente che ha avviato l'agent bridge di protocollo, ma può essere memorizzato ovunque sul file system accessibile dall'agent. Per specificare un'altra posizione, aggiungere l'elemento `<credentialsFile>` al file `ProtocolBridgeProperties.xml`. Ad esempio,

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

Prima di poter utilizzare un agent bridge di protocollo, impostare l'associazione delle credenziali modificando questo file per includere le informazioni su host, utente e credenziali. Per ulteriori informazioni ed esempi, consultare [Formato del file delle credenziali bridge del protocollo](#).

**z/OS** Quando si crea il file `ProtocolBridgeCredentials.xml` su una piattaforma z/OS utilizzando IBM WebSphere MQ 7.5 o versioni precedenti, è necessario impostare una tag file prima di modificare il file. Eseguire il seguente comando per contrassegnare il file come avente contenuto ASCII:

```
chtag -t -c IS08859-1 ProtocolBridgeCredentials.xml
```

**Nota:** **z/OS** Su z/OS, è possibile memorizzare il file delle credenziali del bridge di protocollo in un dataset, in cui il nome del file `.xml` può essere specificato dall'utente.

## Procedura

1. Modificare la linea `<tns:server name="server name">` per modificare il valore dell'attributo `name` nel nome server nel file `ProtocolBridgeProperties.xml`.

Gli agent bridge di protocollo creati per IBM WebSphere MQ File Transfer Edition 7.0.4 e versioni precedenti non hanno un file `ProtocolBridgeProperties.xml` (o uscite utente correlate), quindi per IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 1 e versioni successive al nome server viene assegnato automaticamente il nome host del server. Pertanto, se si utilizza un file `ProtocolBridgeCredentials.xml` aggiornato con voci `<server>`, un nome corrispondente al nome host del server corrisponderà.

È possibile utilizzare l'attributo del pattern per specificare che è stato utilizzato un nome server che contiene caratteri jolly o espressioni regolari. Ad esempio,

```
<tns:server name="serverA*" pattern="wildcard">
```

2. Inserire le informazioni sulle credenziali e sull'ID utente nel file come elementi child di `<tns:server>`.

È possibile inserire uno o più dei seguenti elementi nel file:

- Se il server del file di protocollo è un server FTP, FTPS o SFTP, è possibile utilizzare le password per autenticare l'utente che richiede il trasferimento. Inserire le righe seguenti nel file:

```
<tns:user name="FTE User ID"  
  serverUserId="Server User ID"  
  serverPassword="Server Password">  
</tns:user>
```

Quindi, modificare il valore degli attributi.

- name è un'espressione regolare Java che corrisponde all'ID utente MQMD associato alla richiesta di trasferimento MFT
- serverUserId è il valore passato al server di file del protocollo come ID utente di collegamento. Se l'attributo serverUserId non viene specificato, viene utilizzato invece l'ID utente MQMD associato alla richiesta di trasferimento MFT
- serverPassword è la password associata a serverUserId.

L'attributo name può contenere un'espressione regolare Java . Il credential mapper tenta di associare l'ID utente MQMD della richiesta di trasferimento MFT a questa espressione regolare. L'agent bridge di protocollo tenta di mettere in corrispondenza l'ID utente MQMD con l'espressione regolare nell'attributo del nome degli elementi <tns: user> nell'ordine in cui gli elementi esistono nel file. Quando viene trovata una corrispondenza, l'agent bridge di protocollo non cerca ulteriori corrispondenze. Se viene trovata una corrispondenza, i corrispondenti valori serverUserId e serverPassword vengono passati al server di file del protocollo come ID utente e password di collegamento. Le corrispondenze ID utente MQMD sono sensibili al maiuscolo / minuscolo.

- Se il server di file di protocollo è un server SFTP, è possibile utilizzare chiavi pubbliche e private per autenticare l'utente che richiede il trasferimento. Inserire le seguenti righe nel file e modificare il valore degli attributi. L'elemento <tns: user> può contenere uno o più elementi <tns:privateKey>.

```
<tns:user name="FTE User ID"  
  serverUserId="Server User ID"  
  hostKey="Host Key">  
  <tns:privateKey associationName="association"  
    keyPassword="Private key password">  
    Private key file text  
  </tns:privateKey>  
</tns:user>
```

- name è un'espressione regolare Java che corrisponde all'ID utente MQMD associato alla richiesta di trasferimento MFT
- serverUserId è il valore passato al server di file del protocollo come ID utente di collegamento. Se l'attributo serverUserId non viene specificato, viene utilizzato invece l'ID utente MQMD associato alla richiesta di trasferimento MFT
- hostKey è la chiave prevista restituita dal server durante l'accesso
- key è la chiave privata di serverUserId
- keyPassword è la password per la chiave per generare le chiavi pubbliche
- associationName è un valore utilizzato per l'identificazione a scopo di traccia e registrazione

L'attributo name può contenere un'espressione regolare Java . Il credential mapper tenta di associare l'ID utente MQMD della richiesta di trasferimento MFT a questa espressione regolare. L'agent bridge di protocollo tenta di mettere in corrispondenza l'ID utente MQMD con l'espressione regolare nell'attributo del nome degli elementi <tns: user> nell'ordine in cui gli elementi esistono nel file. Quando viene trovata una corrispondenza, l'agent bridge di protocollo non cerca ulteriori corrispondenze. Se viene trovata una corrispondenza, i corrispondenti valori serverUserId e key vengono utilizzati per autenticare l'utente MFT con il server di file del protocollo. Le corrispondenze ID utente MQMD sono sensibili al maiuscolo / minuscolo.

Per ulteriori informazioni sull'utilizzo delle chiavi private con un agent bridge di protocollo, consultare [“Esempio: come configurare un agent bridge di protocollo per utilizzare le credenziali della chiave privata con un server SFTP UNIX”](#) a pagina 277.

**Nota:** 

Quando la richiesta di trasferimento viene scritta nella coda comandi, l'ID utente MQMD potrebbe essere convertito in maiuscolo se la coda comandi dell'agent di origine si trova su un sistema z/OS o IBM i. Di conseguenza, l'ID utente MQMD per lo stesso utente di origine potrebbe arrivare all'uscita delle credenziali nel caso originale o potrebbe essere convertito in maiuscolo a seconda dell'agent di origine specificato nella richiesta di trasferimento. L'uscita di associazione credenziali predefinita esegue corrispondenze sensibili al maiuscolo / minuscolo rispetto all'ID utente MQMD fornito, che potrebbe essere necessario consentire nel file di associazione.

### ***Associazione delle credenziali per un server di file utilizzando le classi di uscita***

Se non si desidera utilizzare la funzione di associazione credenziali predefinita dell'agente bridge di protocollo, è possibile associare le credenziali utente in Managed File Transfer alle credenziali utente sul server di file scrivendo la propria uscita utente. Se si configurano le uscite utente di associazione credenziali, esse sostituiscono la funzione di associazione credenziali predefinita.

### **Informazioni su questa attività**

Managed File Transfer fornisce un'uscita utente di esempio che esegue l'associazione delle credenziali utente. Per ulteriori informazioni, consultare [“Utilizzo dell'uscita utente delle credenziali del bridge di protocollo di esempio”](#) a pagina 276.

Un'uscita utente per la mappatura delle credenziali del bridge di protocollo deve implementare una delle seguenti interfacce:

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`, che consente a un agent bridge di protocollo di trasferire i file da e verso un server di file di protocollo predefinito
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`, che consente di trasferire file da e verso più endpoint.

L'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` contiene la stessa funzione di `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` e include anche la funzione estesa. Per ulteriori informazioni, vedi [ProtocolBridgeCredentialExit.java](#) [interface](#) e l'interfaccia [ProtocolBridgeCredentialExit2.java](#).

Le uscite delle credenziali possono essere concatenate insieme in modo simile ad altre uscite utente. Le uscite vengono richiamati nell'ordine in cui vengono specificate utilizzando la proprietà `protocolBridgeCredentialConfiguration` del file delle proprietà dell'agente. I metodi di inizializzazione vengono restituiti tutti separatamente e se uno o più restituisce un valore false, l'agent non viene avviato. L'errore viene riportato nella registrazione eventi dell'agent.

Viene restituito un solo risultato complessivo per i metodi ID `mapMQUser` di tutte le uscite, come riportato di seguito:

- Se il metodo restituisce un valore `USER_SUCCESSFULLY_MAPPED` o `USER_DENIED_ACCESS` come codice risultato, questo valore è il risultato restituito e i metodi ID `mapMQUser` delle uscite successive non vengono richiamati.
- Se il metodo restituisce un valore `NO_MAPPING_FOUND` come codice di risultato, viene richiamato il metodo ID `mqMQUser` dell'uscita successiva.
- Se non esiste alcuna uscita successiva, viene restituito il risultato `NO_MAPPING_FOUND`.
- Un codice di risultato generale `USER_DENIED_ACCESS` o `NO_MAPPING_FOUND` viene considerato come un errore di trasferimento dall'agent bridge.

Per eseguire l'uscita, completare la seguente procedura:

## Procedura

1. Compilare la user exit delle credenziali del bridge di protocollo.
2. Creare un file JAR ( Java archive) contenente l'uscita compilata e la relativa struttura del package.
3. Inserire il file JAR che contiene la classe di exit nella directory `exits` dell'agent bridge. La directory si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` .
4. Modificare il file delle proprietà dell'agent bridge di protocollo per includere la proprietà `protocolBridgeCredentialExitClasses`. Per il valore di questa proprietà, specificare un elenco separato da virgole di classi che implementano una routine di uscita delle credenziali del bridge di protocollo. Le classi di uscita vengono richiamati nell'ordine in cui vengono specificate in questo elenco. Per ulteriori informazioni, consultare [Il file MFT agent.properties](#).
5. Modificare il file delle proprietà dell'agent bridge di protocollo per includere:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

Il file `agent.properties` per un agente si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name` .

Se si modifica il file `agent.properties` , è necessario riavviare l'agent per rendere effettive le modifiche.

6. Facoltativamente, è possibile specificare la proprietà `protocolBridgeCredentialConfiguration` . Il valore specificato per questa proprietà viene passato come oggetto String al metodo `initialize ()` delle classi di uscita specificate da `protocolBridgeCredentialExitClasses`. Per ulteriori informazioni, consultare [Il file MFT agent.properties](#).
7. Avviare l'agent bridge di protocollo con il comando **fteStartAgent** .

### Utilizzo dell'uscita utente delle credenziali del bridge di protocollo di esempio

Managed File Transfer fornisce un'uscita utente di esempio che esegue l'associazione delle credenziali utente.

## Informazioni su questa attività

Un'uscita credenziali bridge di protocollo di esempio viene fornita nella directory `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` e nell'argomento [Uscita utente credenziali bridge di protocollo di esempio](#). Questo esempio si basa sull'interfaccia `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` .

L'uscita `SampleCredentialExit.java` legge un file delle proprietà che associa gli ID utente MQMD associati alle richieste di trasferimento agli ID utente del server e alle password del server. L'ubicazione del file delle proprietà viene presa dalla proprietà dell'agent bridge di protocollo `protocolBridgeCredentialConfiguration`.

Per eseguire l'uscita utente di esempio, completare la seguente procedura:

## Procedura

1. Compilare il file `SampleCredentialExit.java` .
2. Creare un file JAR che contenga l'uscita compilata e la relativa struttura del package.
3. Inserire il file JAR nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits` .
4. Modificare il file `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` in modo che contenga la riga:

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. Modificare il file delle proprietà dell'agent bridge di protocollo per includere:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

Il file `agent.properties` per un agente si trova nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name`.

Se si modifica il file `agent.properties`, è necessario riavviare l'agent per rendere effettive le modifiche.

6. Creare un file delle proprietà delle credenziale (`credentials.properties`) nella directory `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` e modificarlo per includere le voci nel formato:

```
mqUserId=serverUserId,serverPassword
```

7. Modificare il file `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` in modo che contenga la riga:

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

È necessario utilizzare il percorso assoluto del file `credentials.properties`.

8. Avviare l'agent bridge di protocollo utilizzando il comando **fteStartAgent**.

## Esempio: come configurare un agent bridge di protocollo per utilizzare le credenziali della chiave privata con un server SFTP UNIX

Questo esempio dimostra come generare e configurare il file `ProtocolBridgeCredentials.xml`. Questo esempio è un esempio tipico e i dettagli possono variare a seconda della piattaforma, ma i principi rimangono gli stessi.

### Informazioni su questa attività

#### Procedura

1. Generare una chiave pubblica e privata da utilizzare per l'autenticazione con il server SFTP.

Ad esempio, su un sistema host Linux, è possibile utilizzare lo strumento **ssh-keygen**, fornito come parte del pacchetto 'openssh', per creare la coppia di chiave pubblica / privata.

Per impostazione assunta, senza argomenti, il comando **ssh-keygen** richiede un'ubicazione e una passphrase per i due file chiave, che per impostazione predefinita sono i nomi:

```
id_rsa      <-- Private key
id_rsa.pub  <-- Public key
```



**Attenzione:** Se si sta utilizzando il comando **ssh-keygen** da una versione recente di OpenSSH, come quella fornita con RHEL 8, il formato chiave utilizzato non è compatibile con l'agent bridge di protocollo e i tentativi di trasferimento al server SFTP hanno esito negativo con il messaggio:

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed
because of invalid private key.
```

Per creare una chiave privata compatibile con queste versioni più recenti di OpenSSH, specifica il formato della chiave con il seguente argomento nel comando **ssh-keygen** :

```
ssh-keygen -m PEM
```

Il contenuto della chiave privata `id_rsa` ha quindi la prima e l'ultima riga di:

```
-----BEGIN RSA PRIVATE KEY-----  
.....  
-----END RSA PRIVATE KEY-----
```

compatibile con l'agent bridge di protocollo.

2. Copiare l'intero contenuto del file `id_rsa.pub` nel file `~/.ssh/authorized_keys` dell'utente SFTP sul server SFTP.

Assicurarsi che le autorizzazioni del file su questo file e la directory `~/.ssh` siano impostati in modo appropriato per il server SFTP per consentire l'autenticazione della chiave. Queste autorizzazioni sono generalmente:

```
~/.ssh          Mode 700  
~/.ssh/authorized_keys  Mode 600
```

3. Managed File Transfer richiede un'impronta digitale ssh host generata utilizzando l'algoritmo MD5 . Eseguire uno dei seguenti comandi per ottenere l'impronta digitale ssh dell'host del server SFTP.

- Per Red Hat® Enterprise Linux versione 6.x e precedenti e Linux Ubuntu 14.04, immetti il seguente comando:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- A cominciare da Red Hat Enterprise Linux 7.x, Linux Ubuntu 16.04 e SuSE Linux 12.4, il comando `ssh-keygen` genera, per impostazione predefinita, l'impronta digitale ssh utilizzando l'algoritmo SHA256 . Per generare l'impronta SSH utilizzando l'algoritmo MD5 , eseguire il seguente comando:

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

L'output del comando sarà simile al seguente esempio:

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

Estrarre la parte esadecimale solo dell'output da utilizzare come `hostKey` nel file `ProtocolBridgeCredentials.xml` (consultare il passo "4" a pagina 278). Pertanto, in questo esempio, è possibile estrarre `64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27`.

4. Sul sistema agent bridge di protocollo, modificare il file `ProtocolBridgeCredentials.xml` . Sostituire i valori mostrati in corsivo nel seguente esempio con i propri valori:

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"  
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials  
ProtocolBridgeCredentials.xsd ">  
  
<tns:agent name="Agent_name">  
  
<tns:server name="SFTP_name">  
  
<tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"  
hostKey="ssh_host_finger">  
<tns:privateKey associationName="name" keyPassword="pass_phrase">  
Complete contents of the id_rsa file including the entries  
-----BEGIN RSA PRIVATE KEY-----  
  
-----END RSA PRIVATE KEY-----  
</tns:privateKey>  
</tns:user>  
</tns:server>
```

```
</tns:agent>
</tns:credentials>
```

dove:

- *nome\_agent* è il nome dell'agent bridge di protocolli.
- *SFTP\_host\_name* è il nome del server SFTP come mostrato nel file `ProtocolBridgeProperties.xml`.
- *mq\_User\_ID* è l'ID utente MQMD associato alla richiesta di trasferimento.
- *SFTP\_user\_ID* è l'ID utente SFTP utilizzato nel passo 2. È il valore passato all'SFTP utilizzato come ID utente di login.
- *ssh\_host\_finger* è l'impronta digitale raccolta nel passo 3.
- *name* è un nome che è possibile specificare per essere utilizzato per scopi di traccia e registrazione.
- *pass\_phrase* è la passphrase che hai fornito nel ssh - keygen nel passo 1.
- *Contenuto completo del file id\_rsa* è il contenuto completo del file `id_rsa` generato dal passo 1. Per prevenire un errore di connessione, assicurarsi di includere entrambe le seguenti voci:

```
-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

È possibile aggiungere ulteriori chiavi duplicando l'elemento `<tns:privatekey>`.

5. Avviare l'agent bridge di protocollo se l'agent non è già avviato. In alternativa, l'agent bridge di protocollo esegue periodicamente il polling del file `ProtocolBridgeCredentials.xml` e apporta le modifiche.

## Configurazione di un bridge di protocollo per un server FTPS

Configurare un server FTPS in modo simile a come si configura un server FTP: creare un agent bridge per il server, definire le proprietà del server e associare le credenziali utente.

### Informazioni su questa attività

Per configurare un server FTPS, completare la seguente procedura:

#### Procedura

1. Creare un agent bridge di protocollo per il server FTPS utilizzando il comando **fteCreateBridgeAgent**. I parametri applicabili a FTP sono applicabili anche a FTPS, ma ci sono anche tre parametri richiesti specifici a FTPS:
  - a) Il parametro **-bt**. Specificare FTPS come valore di questo parametro.
  - b) Il parametro **-bts** per il file truststore. Il comando presuppone che sia richiesta solo l'autenticazione del server ed è necessario specificare l'ubicazione del file truststore.

Il formato esplicito del protocollo FTPS è configurato dal comando **fteCreateBridgeAgent** per impostazione predefinita, ma è possibile configurare il formato implicito modificando il file delle proprietà del bridge di protocollo. Il bridge di protocollo si connette sempre ai server FTPS in modalità passiva.

Per ulteriori informazioni sul comando **fteCreateBridgeAgent**, consultare [fteCreateBridgeAgent \(creare e configurare un agent bridge di protocollo MFT\)](#).

Se sono necessarie istruzioni su come creare i file truststore, consultare l'articolo IBM Developer, [Configuring Secure Sockets Layer connectivity in IBM WebSphere MQ File Transfer Edition](#), oppure consultare le informazioni su keytool nella [documentazione di Oracle keytool](#).

**Nota:** IBM WebSphere MQ File Transfer Edition (FTE) non è più un prodotto supportato. Per eseguire la migrazione da FTE al componente MFT (Managed File Transfer) in IBM MQ, consultare [Migrazione di MFT \(Managed File Transfer\)](#).

2. Definire le proprietà del server FTPS all'interno di un elemento `<ftpsServer>` nel file delle proprietà del bridge di protocollo: `ProtocolBridgeProperties.xml`. Per ulteriori informazioni, fare riferimento a “Definizione delle proprietà per i server di file del protocollo utilizzando il file `ProtocolBridgeProperties.xml`” a pagina 266. È anche possibile abilitare l'autenticazione client modificando il file delle proprietà del bridge di protocollo. Per i dettagli di tutte le opzioni di configurazione, vedere [Formato file delle proprietà del bridge di protocollo](#).
3. Associare le credenziali utente in Managed File Transfer alle credenziali utente sul server FTPS utilizzando la funzione di associazione credenziali predefinita dell'agent bridge di protocollo o scrivendo la propria uscita utente. Per ulteriori informazioni, consultare [Associazione delle credenziali per un server di file](#) a pagina 273.
4. Per impostazione predefinita, il file truststore è configurato per avere il formato JKS; se si desidera modificare il formato, modificare il file delle proprietà del bridge di protocollo.

### Esempio

Di seguito è riportata una voce di esempio per un server FTPS nel file delle proprietà del bridge di protocollo:

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpsserver.mycompany.com" />

  <tns:ftpsServer name="ftpsserver.mycompany.com" host="ftpsserver.mycompany.com" port="990"
  platform="windows"
  timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false"
  trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

### Operazioni successive

Per informazioni relative alle parti del protocollo FTPS supportate e non supportate, consultare [Supporto server FTPS dal bridge di protocollo](#).

## Scenari ed esempi per limitare il numero di trasferimenti di file ai singoli server di file

Come funziona l'agent bridge di protocollo revisionato con gli attributi **maxActiveDestinationTransfers** e **failTransferWhenCapacityReached**, insieme ad alcuni esempi.

### Scenari che mostrano il funzionamento dell'agente bridge di protocollo basato sul valore **maxActiveDestinationTransfers**

#### Scenario 1

Il file `ProtocolBridgeProperties.xml` per un agent bridge di protocollo contiene due definizioni di server di file:

- Non è stato impostato l'attributo **maxActiveDestinationTransfers** globale.
- Non è stato impostato l'attributo **maxActiveDestinationTransfers** su `fileServerA` e `FileServerB`.

- L'attributo **maxDestinationTransfers** dell'agent bridge di protocollo è stato impostato sul valore predefinito.

Se l'attributo **maxDestinationTransfers** dell'agent bridge di protocollo è stato impostato sul valore predefinito di 25,:

- L'agent di destinazione inizia l'elaborazione di due trasferimenti gestiti in fileServerA.
- Entrambi i trasferimenti sono stati completati.

A questo punto, il client si rende conto che fileServerA non è riuscito e imposta i seguenti valori per fileServerA nel file ProtocolBridgeProperties.xml :

**maxActiveDestinationTransfers** = 0  
**failTransferWhenCapacityReached** = vero

- Un altro trasferimento arriva per fileServerA e alcuni per fileServerB:

In base alla proprietà impostata nel passo precedente, il trasferimento gestito a fileServerA viene rifiutato e contrassegnato come non riuscito, mentre i trasferimenti per fileServerB sono gestiti nel flusso esistente standard.

- Dopo qualche tempo, il client scopre che fileServerA è di nuovo in esecuzione, quindi il client rimuove o commenta il valore aggiunto precedentemente in ProtocolBridgeProperties.xml. Arriva un nuovo trasferimento gestito per fileServerA e viene gestito nel flusso esistente standard.

## Scenario 2

- È stato impostato l'attributo **maxActiveDestinationTransfers** per un server di file e non l'attributo **failTransferWhenCapacityReached**.
- L'agent bridge di protocollo agisce come agent di destinazione per questo numero di trasferimenti gestiti al file server.
- Il valore dell'attributo **maxActiveDestinationTransfers** è ridotto di 1.

L'agent bridge di protocollo aggiorna dinamicamente la relativa configurazione e imposta **maxActiveDestinationTransfers** sul nuovo valore mentre è ancora attivo. I trasferimenti gestiti in corso non sono interessati da questo aggiornamento e possono essere completati.

## Scenario 3

Il file ProtocolBridgeProperties.xml per un agent bridge di protocollo contiene due definizioni di file server:

- Non è stato impostato l'attributo **maxActiveDestinationTransfers** globale.
- L'attributo **failTransferWhenCapacityReached** non è stato impostato.
- **maxActiveDestinationTransfers** è stato impostato su 1 su fileServerA.
- L'attributo **maxActiveDestinationTransfers** non è stato impostato su fileServerB.

Se l'agent bridge di protocollo ha l'attributo **maxDestinationTransfers** impostato su 5:

- Il numero massimo di trasferimenti di destinazione attivi dall'agent bridge di protocollo a fileServerA è 1 (sebbene l'agent di destinazione abbia 5 slot di trasferimento di destinazione, solo 1 può essere utilizzato per i trasferimenti gestiti a fileServerA).

Ciò è utile quando fileServerA non riesce. Una volta che fileServerA è di nuovo in esecuzione, il valore di **maxActiveDestinationTransfers** può essere aumentato a 5 per consentire la piena capacità dei trasferimenti di destinazione consentiti.

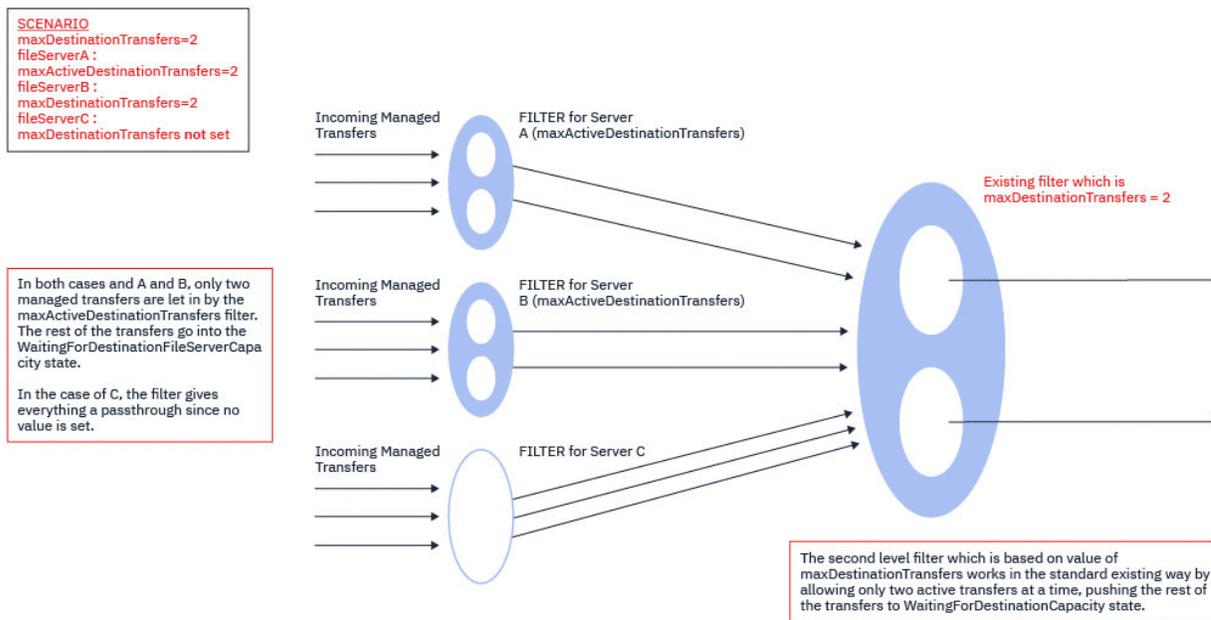
- Il numero massimo di trasferimenti di destinazione attivi dall'agent bridge di protocollo a fileServerB è 5.

Poiché **maxActiveDestinationTransfers** non è impostato per questo server di file, l'agent bridge di protocollo può utilizzare tutti i 5 slot di trasferimento di destinazione per i trasferimenti gestiti.

## Scenario 4

Nel seguente diagramma:

- Si è impostato l'attributo **maxDestinationTransfers** su 2 nel file `agent.properties`.
- Il **maxActiveDestinationTransfers** è stato impostato su 2 su `fileServerA`.
- L'attributo **maxActiveDestinationTransfers** è stato impostato su 2 su `fileServerB`.
- L'attributo **maxActiveDestinationTransfers** non è stato impostato su `fileServerC`.



Come mostra il diagramma, gli attributi **maxActiveDestinationTransfers** e **maxDestinationTransfers** sono indipendenti l'uno dall'altro.

I valori per **maxActiveDestinationTransfers** per ciascuno dei server vengono controllati. In base a questo valore, ai trasferimenti è consentito continuare o passare allo stato **WaitingForDestinationFileServerCapacity**.

I trasferimenti consentiti passano quindi attraverso il flusso standard esistente di verifica rispetto al **maxDestinationTransfers**.

## Scenario 5



**Attenzione:** Prestare attenzione quando si impostano i valori degli attributi **maxActiveDestinationTransfers**, poiché è necessario tenere presente il valore dell'attributo **maxDestinationTransfers**.

Se non si esegue questa operazione, può verificarsi una situazione come descritto nel testo seguente:

- Non è stato impostato un valore per l'attributo **maxActiveDestinationTransfers** globale.
- È stato impostato il valore **maxDestinationTransfers=2** nel file `agent.properties`.
- È stato impostato il valore **maxActiveDestinationTransfers=2** su `fileServerA`.
- Non è stato impostato un valore per **maxActiveDestinationTransfers** su `fileServerB`.

Si supponga che si verifichi la seguente sequenza di eventi:

- L'agent bridge di protocollo riceve una richiesta di trasferimento di un file a `fileServerA`. L'agent bridge di protocollo non sta attualmente eseguendo alcuna operazione, quindi accetta questa richiesta di trasferimento gestito.

Gli slot di trasferimento ora hanno il seguente aspetto:

- Trasferimenti di destinazione: 1
- Trasferimenti di destinazione per fileServerA: 1
- Trasferimenti di destinazione per fileServerB: 0
- A questo punto, l'agent bridge di protocollo riceve un'ulteriore richiesta di agire come agent di destinazione per un trasferimento gestito che coinvolge fileServerA. Ancora una volta, accetta questa richiesta e quindi gli slot di trasferimento si assomigliano a questo:
  - Trasferimenti destinazione: 2
  - Trasferimenti di destinazione per fileServerA: 2
  - Trasferimenti di destinazione per fileServerB: 0

I due slot Destination Transfer nell'agent sono ora occupati e quindi l'agent non può partecipare a ulteriori trasferimenti gestiti fino a quando uno dei trasferimenti a fileServerA non è terminato.

- Un breve periodo di tempo dopo, fileServerA non riesce, causando il ripristino dei due trasferimenti gestiti. Gli slot Destination Transfer che questi trasferimenti gestiti utilizzano restano in uso durante questo periodo di tempo.
- Successivamente, l'agente bridge di protocollo riceve una richiesta di trasferimento di un file a fileServerB. C'è uno spazio per questo trasferimento negli slot Destination Transfers for fileServerB, tuttavia, vengono utilizzati tutti gli slot Destination Transfer per l'agent e quindi il trasferimento viene inserito nel backlog in modo che possa essere ritentato in un secondo momento.

Di conseguenza, il trasferimento a fileServerB viene bloccato fino a che almeno uno dei trasferimenti a fileServerA non ha completato e rilasciato il suo slot Destination Transfer.

Per evitare che questa situazione si verifichi:

- Impostare il valore di **maxActiveDestinationTransfers** sui server di file in modo che sia inferiore al valore **maxDestinationTransfers**, in modo che gli slot liberi rimangano o
- Distribuire uniformemente il valore dell'attributo **maxActiveDestinationTransfers** tra tutti i server endpoint.

## Funzionamento dell'agent bridge di protocollo basato sui valori dell'attributo **maxActiveDestinationTransfers**

**Nota:** In tutti i casi di errore elencati nella seguente tabella, se l'attributo **maxActiveDestinationTransfers** è impostato su un valore non valido, l'agent bridge di protocollo presume che questo attributo non sia impostato.

<b>maxActiveDestinationTransfers</b>	<b>Valore di esempio</b>	<b>Descrizione</b>
Non specificato	Non specificato	I trasferimenti vanno come al solito. Non esiste alcun limite al numero di trasferimenti per l'endpoint * ftp *.
Specificata	0	Nessun trasferimento consentito a questo endpoint * ftp * specifico.
Valore negativo	-1	Errore registrato in output0.log Value -1 non è valido per un numero intero non negativo. L'agent bridge di protocollo presuppone che l'attributo non sia impostato.
Valore non intero	abc	Errore registrato in output0.log Valore abc non valido per un valore intero.

maxActiveDestinationTransfers	Valore di esempio	Descrizione
		L'agent bridge di protocollo presuppone che l'attributo non sia impostato.
Vuoto	""	Il valore '' dell'attributo maxActiveDestinationTransfers non è valido per un intero non negativo.
Specificata	5	Consente di eseguire solo cinque trasferimenti attivi in qualsiasi momento per questo endpoint * ftp *.  I trasferimenti eccessivi vengono ritentati o rifiutati, in base al valore dell'attributo failTransferWhenCapacityReached.

### Funzionamento dell'agent bridge di protocollo per la combinazione di attributi maxActiveDestinationTransfers e failTransferWhenCapacityReached

failTransferWhenCapacityValue e raggiunto	Valore maxActiveDestinationTransfers	Risultato
Falso	3	Sono consentiti tre trasferimenti attivi a questo server endpoint. Qualsiasi altro trasferimento viene ritentato.
Vero	3	Sono consentiti tre trasferimenti attivi a questo server endpoint. Eventuali altri trasferimenti vengono rifiutati e contrassegnati come non riusciti.
Non specificato	3	Viene considerato il valore predefinito <b>false</b> per failTransferWhenCapacityReached.  Il risultato è che tre trasferimenti attivi sono consentiti a questo server endpoint. Qualsiasi altro trasferimento viene ritentato.
Valori diversi dal valore booleano	Specificata	Errore registrato in output.log.  Il valore specificato per failTransferWhenCapacityRaggiunto non è un valore booleano.  Viene considerato il valore predefinito per failTransferWhenCapacityReached.

## Funzionamento dell'agent bridge di protocollo per la combinazione di attributi `maxDestinationTransfers` e `failTransferWhenCapacityReached`

<code>failTransferWhenCapacityValor</code> e raggiunto	<code>maxDestinationValore</code> trasferimenti	Risultato
Vero	10	Quando il numero di trasferimenti attivi simultanei raggiunge 10, l'11 <sup>esimo</sup> trasferimento gestito non riesce dall'agent bridge di protocollo.
Falso	10	Comportamento esistente. Quando il numero di trasferimenti attivi simultanei raggiunge 10, l'11 <sup>esimo</sup> trasferimento gestito viene accodato in attesa che venga liberato uno slot.
Non specificato	10	Comportamento esistente

### Messaggi di errore

Messaggio esistente:

#### **BFGS0082I**

Viene registrato nel file `output0.log` dell'agent di origine quando l'agent bridge di protocollo rifiuta il trasferimento, quando l'agent bridge di protocollo sta già eseguendo il numero massimo di trasferimenti definiti nell'attributo `maxDestinationTransfers`.

Nuovi messaggi:

#### **BFGSS0085I**

Viene registrato nel file `output0.log` dell'agent di origine quando l'agent bridge di protocollo rifiuta e ritenta un trasferimento gestito,

#### **BFGSS0086I**

Viene registrato nel file `output0.log` dell'agent di origine quando l'agent bridge di protocollo rifiuta e ritenta un trasferimento gestito e l'item di destinazione non include il nome del server di file

#### **BFGSS0084E**

Viene registrato nel file `Explorer` e `audit.xml` quando l'agent bridge di protocollo rifiuta, per il superamento del numero massimo di trasferimenti simultanei specificato nell'attributo `maxActiveDestinationTransfers` e contrassegna un trasferimento gestito come non riuscito.

#### **BFGSS0087E**

Viene registrato nel file `Explorer` e `audit.xml` quando l'agent bridge di protocollo rifiuta, per aver superato il numero massimo di trasferimenti di destinazione specificati nell'attributo `maxActiveDestinationTransfers` e contrassegna un trasferimento gestito come non riuscito.

#### **BFGSS0088W**

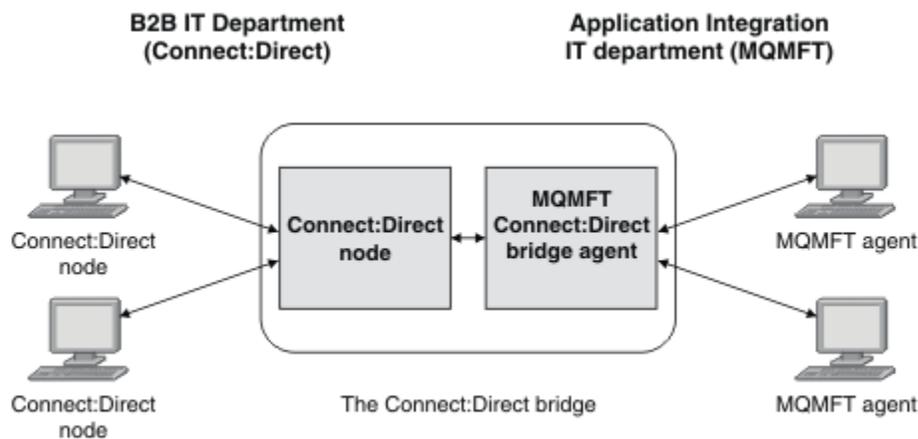
Viene registrato in `output0.log`, quando il valore dell'attributo `maxActiveDestinationTransfers` supera il valore dell'attributo `maxDestinationTransfers`.

#### **BFGSS0089I**

viene registrato nel file `output0.log` dell'agent bridge del protocollo di destinazione, quando sta utilizzando un agent di origine che non si trova in IBM MQ 9.2.1o in una versione successiva.

## Il bridge `Connect:Direct`

È possibile trasferire i file a e da una rete IBM Sterling `Connect:Direct` esistente. Utilizzare il bridge `Connect:Direct`, che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling `Connect:Direct`.



Il diagramma mostra un ponte MFT Connect:Direct tra due dipartimenti, il dipartimento IT B2B e il reparto IT di Application Integration. Il reparto IT B2B usa Connect:Direct per trasferire i file da e verso i business partner della società. Il dipartimento IT di Application Integration utilizza IBM MQ come infrastruttura di messaggistica e ha scelto di recente Managed File Transfer come soluzione di trasferimento file.

Utilizzando il bridge MFT Connect:Direct, i due dipartimenti possono trasferire file tra la rete Connect:Direct nel dipartimento IT B2B e la rete MFT nel dipartimento IT di Application Integration. Il bridge di Connect:Direct è un componente di Managed File Transfer, che include un agente MFT che comunica con un nodo Connect:Direct. L'agente MFT è dedicato ai trasferimenti con il nodo Connect:Direct ed è noto come agent bridge Connect:Direct.

Il bridge Connect:Direct è disponibile come componente dei componenti Service e Agent di Managed File Transfer e può essere utilizzato per le seguenti attività:

1. Utilizzare i comandi Managed File Transfer per avviare un trasferimento di un file, o di più file, da un agente MFT a un nodo Connect:Direct.
2. Utilizzare i comandi Managed File Transfer per avviare un trasferimento di un file, o più file, da un nodo Connect:Direct a un agente MFT.
3. Utilizzare i comandi Managed File Transfer per avviare un trasferimento file che avvia un processo definito dall'utente Connect:Direct.
4. Utilizzare il processo Connect:Direct per inoltrare una richiesta di trasferimento file MFT.

Un bridge Connect:Direct può trasferire i file solo verso o da nodi Connect:Direct. Il bridge Connect:Direct può trasferire i file da o verso il proprio file system locale solo come parte di un trasferimento inoltrato da un processo Connect:Direct.

► **z/OS** È possibile utilizzare il bridge Connect:Direct per il trasferimento a o da un dataset che si trova su un nodo Connect:Direct su un sistema z/OS. Ci sono alcune differenze nel comportamento rispetto ai trasferimenti di dataset che coinvolgono solo gli agenti Managed File Transfer. Per ulteriori informazioni, consultare ► **z/OS** [Trasferimento di dataset da e verso nodi Connect:Direct.](#)

## Piattaforme supportate

Il bridge Connect:Direct è costituito da un agente bridge MFT Connect:Direct e da un nodo Connect:Direct. L'agente è supportato su Windows e Linux per x86-64. Il nodo è supportato sulle piattaforme supportate per IBM Sterling Connect:Direct per Windows e IBM Sterling Connect:Direct per UNIX. Per istruzioni sulla creazione dell'agente bridge Connect:Direct e sulla configurazione di un nodo Connect:Direct con cui l'agente deve comunicare, consultare [Configurazione del bridge Connect:Direct.](#)

Il bridge Connect:Direct può trasferire i file da e verso i nodi Connect:Direct in esecuzione come parte di un'installazione del servizio Connect:Direct per Windows o Connect:Direct per UNIX ► **z/OS** o

Connect:Direct per z/OS . Per dettagli sulle versioni di Connect:Direct supportate, consultare la pagina [Web Requisiti di sistema per IBM MQ](#).

L'agent e il nodo che costituiscono il bridge Connect:Direct devono essere sullo stesso sistema o avere accesso allo stesso file system, ad esempio tramite un montaggio NFS condiviso. Questo file system viene utilizzato per memorizzare temporaneamente i file durante i trasferimenti di file che coinvolgono il bridge Connect:Direct, in una directory definita dal parametro **cdTmpDir**. L'agent bridge Connect:Direct e il nodo bridge Connect:Direct devono essere in grado di raggiungere questa directory utilizzando lo stesso nome percorso. Ad esempio, se l'agent e il nodo si trovano su sistemi Windows separati, i sistemi devono utilizzare la stessa lettera di unità per il montaggio del file system condiviso. Le seguenti configurazioni consentono all'agent e al nodo di utilizzare lo stesso nome percorso:

- L'agent e il nodo si trovano sullo stesso sistema, che è in esecuzione in Windows o Linux per x86-64
- L'agent si trova su Linux per x86-64 e il nodo è su AIX
- L'agent si trova su un sistema Windows e il nodo si trova su un altro sistema Windows

Le seguenti configurazioni non consentono all'agent e al nodo di utilizzare lo stesso nome percorso:

- L'agent si trova su Linux per x86-64 e il nodo è su Windows
- L'agent si trova su Windows e il nodo su UNIX

Considerare questa limitazione quando si pianifica l'installazione del bridge Connect:Direct.

## Trasferimento di un file in un nodo Connect:Direct

È possibile trasferire un file da un agent Managed File Transfer a un nodo Connect:Direct utilizzando il bridge Connect:Direct . Specifica un nodo Connect:Direct come destinazione del trasferimento specificando l'agent bridge Connect:Direct come agent di destinazione e specificando il file di destinazione nel formato `connect_direct_node_name:file_path`.

### Prima di iniziare

Prima di trasferire un file, è necessario configurare il bridge Connect:Direct , che è un componente di Managed File Transfer. Per ulteriori informazioni, consultare [Configurazione del bridge Connect:Direct](#).

### Informazioni su questa attività

In questo esempio, l'agent bridge Connect:Direct è denominato CD\_BRIDGE. L'agent di origine è denominato FTE\_AGENT e può essere qualsiasi versione di WMQFTE. Il nodo di destinazione Connect:Direct è denominato CD\_NODE1. Il file da trasferire si trova nel percorso file `/home/helen/file.log` sul sistema in cui si trova FTE\_AGENT. Il file viene trasferito al percorso file `/files/data.log` sul sistema su cui è in esecuzione CD\_NODE1 .

### Procedura

1. Utilizzare il comando `fteCreateTransfer` con il valore per il parametro **-df** (file di destinazione) nel formato `connect_direct_node_name:file_path` e il valore del parametro **-da** (agent di destinazione) specificato come nome dell'agent bridge Connect:Direct .

**Nota:** Il nodo Connect:Direct specificato da `connect_direct_node_name` è il nodo in cui si desidera trasferire il file, non quello Connect:Direct che opera come parte del bridge Connect:Direct .

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                 -df CD_NODE1:/files/data.log /home/helen/file.log
```

Per ulteriori informazioni, consultare [fteCreateTransfer: avviare un nuovo trasferimento file](#).

2. L'agent di origine FTE\_AGENT trasferisce il file all'agent bridge Connect:Direct CD\_BRIDGE. Il file viene memorizzato temporaneamente sul sistema su cui è in esecuzione l'agent bridge Connect:Direct , nell'ubicazione definita dalla proprietà dell'agent Dir `cdTmp`. L'agent bridge Connect:Direct trasferisce il file al nodo Connect:Direct CD\_NODE1.

## Concetti correlati

[“Il bridge Connect:Direct” a pagina 285](#)

È possibile trasferire i file a e da una rete IBM Sterling Connect:Direct esistente. Utilizzare il bridge Connect:Direct , che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling Connect:Direct.

## Attività correlate

[“Trasferimento di un file da un nodo Connect:Direct” a pagina 288](#)

Puoi trasferire un file da un nodo Connect:Direct a un Managed File Transfer Agent utilizzando il bridge Connect:Direct . È possibile specificare un nodo di Connect:Direct come origine del trasferimento specificando l'agent bridge Connect:Direct come agent di origine e specificando la specifica di origine nel formato `connect_direct_node_name:file_path`.

## Riferimenti correlati

[Il file MFT agent.properties](#)

## Trasferimento di un file da un nodo Connect:Direct

Puoi trasferire un file da un nodo Connect:Direct a un Managed File Transfer Agent utilizzando il bridge Connect:Direct . È possibile specificare un nodo di Connect:Direct come origine del trasferimento specificando l'agent bridge Connect:Direct come agent di origine e specificando la specifica di origine nel formato `connect_direct_node_name:file_path`.

## Prima di iniziare

Prima di trasferire un file, è necessario configurare il bridge Connect:Direct , che è un componente di Managed File Transfer. Consultare [Configurazione del bridge Connect:Direct](#).

## Informazioni su questa attività

In questo esempio, l'agent bridge Connect:Direct è denominato CD\_BRIDGE. L'agent di destinazione è denominato FTE\_AGENT e può essere una qualsiasi versione di Managed File Transfer. Il nodo Connect:Direct di origine è denominato CD\_NODE1. Il file da trasferire si trova nel percorso file `/home/brian/in.file` sul sistema in cui si trova CD\_NODE1 . Il file viene trasferito al percorso file `/files/out.file` sul sistema su cui è in esecuzione FTE\_AGENT.

## Procedura

Utilizzare il comando **fteCreateTransfer** con il valore per la specifica di origine nel formato `connect_direct_node_name:file_path` e il valore del parametro **-sa** specificato come nome dell'agente bridge Connect:Direct .

**Nota:** Il nodo Connect:Direct , specificato da `connect_direct_node_name` , è il nodo da cui si desidera trasferire il file e non il nodo Connect:Direct che opera come parte del bridge Connect:Direct . Ad esempio:

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                  -df /files/out.file CD_NODE1:/home/brian/in.file
```

Per ulteriori informazioni, consultare [fteCreateTransfer: avviare un nuovo trasferimento file](#).

## Risultati

L'agent bridge Connect:Direct CD\_BRIDGE richiede il file dal nodo Connect:Direct CD\_NODE1. Il nodo Connect:Direct invia il file al bridge Connect:Direct . Durante il trasferimento del file dal nodo Connect:Direct , il bridge Connect:Direct memorizza il file temporaneamente nell'ubicazione definita dalla proprietà dell'agent `cdTmpDir` . Una volta terminato il trasferimento dal nodo Connect:Direct al bridge Connect:Direct , il bridge Connect:Direct invia il file all'agent di destinazione FTE\_AGENT ed elimina il file dall'ubicazione temporanea.

## Concetti correlati

[“Il bridge Connect:Direct” a pagina 285](#)

È possibile trasferire i file a e da una rete IBM Sterling Connect:Direct esistente. Utilizzare il bridge Connect:Direct , che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling Connect:Direct.

### Riferimenti correlati

[Il file MFT agent.properties](#)

## Trasferimento di un dataset a un nodo Connect:Direct su z/OS

È possibile trasferire un dataset da un agent Managed File Transfer su z/OS a un nodo Connect:Direct su z/OS utilizzando un bridge Connect:Direct ubicato su un sistema Windows o Linux .

### Prima di iniziare

Prima di trasferire un file, è necessario configurare il bridge Connect:Direct , che è un componente di Managed File Transfer. Consultare [Configurazione del bridge Connect:Direct](#).

### Informazioni su questa attività

In questo esempio, il parametro **-df** viene utilizzato per specificare la destinazione del trasferimento. Il parametro **-df** è valido per l'utilizzo quando l'agent di origine del trasferimento è una qualsiasi versione di Managed File Transfer. Se l'agent di origine è IBM WebSphere MQ File Transfer Edition 7.0.4 o successivo, è possibile utilizzare il parametro **-ds** . L'agent di origine è denominato FTE\_ZOS1 ed è un agent IBM WebSphere MQ File Transfer Edition 7.0.3 . L'agent bridge Connect:Direct è denominato CD\_BRIDGE e si trova su un sistema Linux . Il nodo di destinazione Connect:Direct è denominato CD\_ZOS2. Sia l'agent di origine che il nodo Connect:Direct di destinazione si trovano su sistemi z/OS . Il dataset da trasferire si trova in //FTEUSER.SOURCE.LIB sul sistema in cui si trova FTE\_ZOS1 . Il dataset viene trasferito al dataset //CDUSER.DEST.LIB sul sistema in cui si trova CD\_ZOS2 .

**Nota:** IBM WebSphere MQ File Transfer Edition (FTE) non è più un prodotto supportato. Per eseguire la migrazione da FTE al componente MFT (Managed File Transfer) in IBM MQ, consultare [Migrazione di MFT \(Managed File Transfer\)](#).

### Procedura

1. Utilizzare il comando di trasferimento `fteCreateTransfer` con il valore per il parametro **-df** nel formato: `connect_direct_node_name:data_set_name;attributes` e il valore del parametro **-da** (agent di destinazione) specificato come nome dell'agent bridge Connect:Direct .

Il nodo Connect:Direct specificato da `connect_direct_node_name` è il nodo a cui si desidera trasferire il dataset, non il nodo Connect:Direct che opera come parte del bridge Connect:Direct .

Il nome dataset specificato da `nome_set_dati` deve essere assoluto, non relativo. Connect:Direct non antepone il nome del dataset al nome dell'utente.

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                 -da CD_BRIDGE -dm QM_BRIDGE
                 -df CD_ZOS2://'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80) '
                 //'FTEUSER.SOURCE.LIB'
```

Per ulteriori informazioni, consultare [fteCreateTransfer: avviare un nuovo trasferimento file](#).

2. L'agent di origine FTE\_ZOS1 trasferisce i dati nel dataset all'agent bridge Connect:Direct CD\_BRIDGE. I dati vengono memorizzati temporaneamente come file flat sul sistema su cui è in esecuzione l'agente bridge Connect:Direct , nell'ubicazione definita dalla proprietà dell'agente Dir cdTmp. L'agent bridge Connect:Direct trasferisce i dati al nodo Connect:Direct CD\_ZOS2. Una volta completato il trasferimento, il file flat viene eliminato dal sistema su cui è in esecuzione l'agent bridge Connect:Direct .

### Concetti correlati

[“Il bridge Connect:Direct” a pagina 285](#)

È possibile trasferire i file a e da una rete IBM Sterling Connect:Direct esistente. Utilizzare il bridge Connect:Direct , che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling Connect:Direct.

### Attività correlate

 [Trasferimento di dataset da e verso nodi Connect:Direct](#)

### Riferimenti correlati

 [Proprietà BPXWDYN che non devono essere utilizzate con MFT](#)

## Trasferimento di più file in un nodo Connect:Direct

È possibile trasferire più file da un Managed File Transfer Agent ad un nodo Connect:Direct utilizzando il bridge Connect:Direct . Per utilizzare un nodo Connect:Direct come destinazione del trasferimento file multiplo, specificare l'agent bridge Connect:Direct come agent di destinazione e specificare la directory di destinazione nel formato *connect\_direct\_node\_name:directory\_path*.

### Prima di iniziare

Prima di trasferire i file, configurare il bridge Connect:Direct , che è un componente di Managed File Transfer. Consultare [Configurazione del bridge Connect:Direct](#).

### Informazioni su questa attività

In questo esempio, l'agent di origine è denominato FTE\_AGENT. L'agent bridge Connect:Direct è denominato CD\_BRIDGE. Il nodo di destinazione Connect:Direct è denominato CD\_NODE1. I file da trasferire sono */home/jack/data.log*, */logs/log1.txt* e */results/latest* sul sistema in cui si trova FTE\_AGENT. I file vengono trasferiti alla directory */in/files* sul sistema su cui è in esecuzione CD\_NODE1 .

### Procedura

Utilizzare il comando di trasferimento `fteCreate` con il valore per il parametro **-dd** (directory di destinazione) nel formato *connect\_direct\_node\_name:directory\_path*. Specificare il valore del parametro **-da** (agent di destinazione) come nome dell'agent bridge Connect:Direct .

**Nota:** Il nodo Connect:Direct specificato da *connect\_direct\_node\_name* è il nodo in cui si desidera trasferire i file, non il nodo Connect:Direct che opera come parte del bridge Connect:Direct .

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

Per ulteriori informazioni, consultare [fteCreateTransfer](#): avviare un nuovo trasferimento file.

### Risultati

L'agent di origine FTE\_AGENT trasferisce il primo file all'agent bridge Connect:Direct CD\_BRIDGE. L'agent bridge Connect:Direct memorizza temporaneamente il file nell'ubicazione definita dalla proprietà `cdTmpDir`. Quando il file è stato completamente trasferito dall'agent di origine al bridge Connect:Direct , l'agent bridge Connect:Direct invia il file al nodo Connect:Direct definito dalla proprietà dell'agent `cdNode` . Questo nodo invia il file al nodo di destinazione Connect:Direct CD\_NODE1. L'agent bridge Connect:Direct elimina il file dall'ubicazione temporanea quando il trasferimento tra i due nodi Connect:Direct è completo. Questo processo viene ripetuto per ogni file origine specificato.

### Concetti correlati

[“Il bridge Connect:Direct” a pagina 285](#)

È possibile trasferire i file a e da una rete IBM Sterling Connect:Direct esistente. Utilizzare il bridge Connect:Direct , che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling Connect:Direct.

## Attività correlate

[“Trasferimento di un file in un nodo Connect:Direct” a pagina 287](#)

È possibile trasferire un file da un agent Managed File Transfer a un nodo Connect:Direct utilizzando il bridge Connect:Direct . Specifica un nodo Connect:Direct come destinazione del trasferimento specificando l'agent bridge Connect:Direct come agent di destinazione e specificando il file di destinazione nel formato `connect_direct_node_name:file_path`.

[“Trasferimento di più file a Connect:Direct utilizzando caratteri jolly” a pagina 292](#)

Per trasferire più file da un agent Managed File Transfer a un nodo Connect:Direct , utilizzare il bridge Connect:Direct . È possibile utilizzare i caratteri jolly nella specifica di origine fornita al comando **fteCreateTransfer** . Come per tutti i trasferimenti Managed File Transfer che coinvolgono caratteri jolly, solo l'ultima parte del percorso file può contenere un carattere jolly. Ad esempio, `/abc/def*` è un percorso file valido e `/abc*/def` non è valido.

[“Trasferimento di un file da un nodo Connect:Direct” a pagina 288](#)

Puoi trasferire un file da un nodo Connect:Direct a un Managed File Transfer Agent utilizzando il bridge Connect:Direct . È possibile specificare un nodo di Connect:Direct come origine del trasferimento specificando l'agent bridge Connect:Direct come agent di origine e specificando la specifica di origine nel formato `connect_direct_node_name:file_path`.

[“Trasferimento di più file da un nodo Connect:Direct” a pagina 291](#)

È possibile trasferire più file da un nodo Connect:Direct a un Managed File Transfer Agent utilizzando il bridge Connect:Direct . È possibile specificare un nodo Connect:Direct come origine del trasferimento di più file specificando l'agent bridge Connect:Direct come agent di origine e specificando una o più specifiche di origine nel modulo `connect_direct_node_name:file_path`.

## Riferimenti correlati

[Il file MFT agent.properties](#)

## Trasferimento di più file da un nodo Connect:Direct

È possibile trasferire più file da un nodo Connect:Direct a un Managed File Transfer Agent utilizzando il bridge Connect:Direct . È possibile specificare un nodo Connect:Direct come origine del trasferimento di più file specificando l'agent bridge Connect:Direct come agent di origine e specificando una o più specifiche di origine nel modulo `connect_direct_node_name:file_path`.

## Prima di iniziare

Prima di trasferire un file, è necessario configurare il bridge Connect:Direct , che è un componente di Managed File Transfer. Consultare la sezione [Configurazione del bridge Connect:Direct](#).

## Informazioni su questa attività

In questo esempio, l'agent bridge Connect:Direct è denominato CD\_BRIDGE. L'agent di destinazione è denominato FTE\_Z ed è in esecuzione su un sistema z/OS . Il nodo Connect:Direct di origine è denominato CD\_NODE1. I file da trasferire si trovano nei percorsi file `/in/file1`, `/in/file2` e `/in/file3` sul sistema in cui si trova CD\_NODE1 . I file vengono trasferiti al dataset partizionato `//OBJECT.LIB` sul sistema su cui è in esecuzione FTE\_Z.

## Procedura

Utilizzare il comando `fteCreateTransfer` con i valori per le specifiche di origine nel formato `connect_direct_node_name:file_path` e il valore del parametro `-sa` specificato come nome dell'agent bridge Connect:Direct .

**Nota:** Il nodo Connect:Direct specificato da `connect_direct_node_name` è il nodo da cui si desidera trasferire i file, non il nodo Connect:Direct che opera come parte del bridge Connect:Direct .

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp '//OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

Per ulteriori informazioni, consultare [fteCreateTransfer](#): avviare un nuovo trasferimento file.

## Risultati

L'agent bridge Connect:Direct CD\_BRIDGE richiede il primo file dal Connect:Direct nodo CD\_NODE1. Il nodo Connect:Direct invia il file al bridge Connect:Direct. Durante il trasferimento del file dal nodo Connect:Direct, il bridge Connect:Direct memorizza il file temporaneamente nell'ubicazione definita dalla proprietà dell'agent `cdTmpDir`. Una volta terminato il trasferimento del file dal nodo Connect:Direct al bridge Connect:Direct, il bridge Connect:Direct invia il file all'agent di destinazione FTE\_Z ed elimina il file dall'ubicazione temporanea. Questo processo viene ripetuto per ogni file origine specificato.

### Concetti correlati

“Il bridge Connect:Direct” a pagina 285

È possibile trasferire i file a e da una rete IBM Sterling Connect:Direct esistente. Utilizzare il bridge Connect:Direct, che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling Connect:Direct.

### Riferimenti correlati

[Il file MFT agent.properties](#)

## Trasferimento di più file a Connect:Direct utilizzando caratteri jolly

Per trasferire più file da un agent Managed File Transfer a un nodo Connect:Direct, utilizzare il bridge Connect:Direct. È possibile utilizzare i caratteri jolly nella specifica di origine fornita al comando **fteCreateTransfer**. Come per tutti i trasferimenti Managed File Transfer che coinvolgono caratteri jolly, solo l'ultima parte del percorso file può contenere un carattere jolly. Ad esempio, `/abc/def*` è un percorso file valido e `/abc*/def` non è valido.

## Prima di iniziare

Prima di trasferire un file, è necessario configurare il bridge Connect:Direct, che è un componente di Managed File Transfer. Per ulteriori informazioni, consultare [Configurazione del bridge Connect:Direct](#).

## Informazioni su questa attività

In questo esempio, l'agent di origine è denominato FTE\_AGENT e l'agent bridge Connect:Direct è denominato CD\_BRIDGE. Il nodo di destinazione Connect:Direct è denominato CD\_NODE1. I file da trasferire si trovano nella directory `/reports` sul sistema in cui si trova FTE\_AGENT. Vengono trasferiti solo i file con nomi che iniziano con `report`, seguiti da due caratteri e dal suffisso `.log`. Ad esempio, il file `/reports/report01.log` viene trasferito, ma il file `/reports/report1.log` non viene trasferito. I file vengono trasferiti alla directory `/home/fred` sul sistema su cui è in esecuzione CD\_NODE1.

## Procedura

1. Utilizzare il comando di trasferimento `fteCreateTransfer` con il valore per il parametro **-dd** (directory di destinazione) nel formato `connect_direct_node_name:directory_path`. Per il parametro **-da** (agent di destinazione), specificare l'agent bridge Connect:Direct.

**Nota:** Il nodo Connect:Direct specificato da `connect_direct_node_name` è il nodo in cui si desidera trasferire i file, non il nodo Connect:Direct che opera come parte del bridge Connect:Direct.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE  
-dd CD_NODE1:/home/fred "/reports/report?.log"
```

Per ulteriori informazioni, consultare [fteCreateTransfer](#): avviare un nuovo trasferimento file.

2. L'agent di origine FTE\_AGENT trasferisce il primo file che corrisponde al modello `/reports/report?.log` all'agent bridge Connect:Direct CD\_BRIDGE. L'agent bridge Connect:Direct memorizza temporaneamente il file nell'ubicazione definita dalla proprietà `cdTmpDir`. Quando il file è stato completamente trasferito dall'agent di origine al bridge Connect:Direct, l'agent bridge Connect:Direct invia il file al nodo Connect:Direct definito dalla proprietà dell'agent `cdNode`. Questo nodo invia il file al nodo di destinazione Connect:Direct CD\_NODE1. L'agent bridge Connect:Direct elimina il file dall'ubicazione temporanea quando il trasferimento tra i due nodi Connect:Direct è completo. Questo

processo è ripetuto per ogni file di origine che corrisponde al modello carattere jolly `/reports/report??.log`.

**Nota:** L'elenco di file che corrispondono al modello `/reports/report??.log` varia a seconda del sistema operativo del sistema in cui si trova l'agent di origine FTE\_AGENT.

- Se l'agent di origine si trova su un sistema con un sistema operativo Windows, la corrispondenza del modello non è sensibile al maiuscolo / minuscolo. Il modello corrisponde a tutti i file nella directory `/reports` con un nome file in formato `report` seguito da due caratteri e un suffisso `.log`, indipendentemente dal caso in cui si trovano le lettere. Ad esempio, `Report99.Log` è una corrispondenza.
- Se l'agent di origine si trova su un sistema con un sistema operativo Linux o UNIX, la corrispondenza del modello è sensibile al maiuscolo / minuscolo. Il modello corrisponde solo a quei file nella directory `/reports` con un nome file nel formato `report` seguito da due caratteri e un suffisso `.log`. Ad esempio, `reportAB.log` è una corrispondenza, ma `reportAB.LOG` e `Report99.Log` non sono corrispondenze.

## Ripristino e riavvio per i trasferimenti da e verso i nodi Connect:Direct

Managed File Transfer potrebbe non essere in grado di connettersi al tuo nodo IBM Sterling Connect:Direct durante un trasferimento; ad esempio, se il nodo diventa non disponibile. Managed File Transfer tenta di ripristinare il trasferimento o il trasferimento ha esito negativo e viene prodotto un messaggio di errore.

### Se il nodo Connect:Direct diventa non disponibile

Se il nodo Connect:Direct diventa non disponibile; ad esempio, a causa di un'interruzione della rete o dell'alimentazione, Managed File Transfer recupera un trasferimento file nei seguenti modi:

- Se Managed File Transfer non è stato precedentemente connesso correttamente al nodo Connect:Direct come parte di questa richiesta di trasferimento, il trasferimento viene ritentato per un periodo di tempo determinato dai valori di **cdMaxConnectionRetries** e **recoverableTransferRetryInterval properties**. Queste proprietà sono specificate nel file `agent.properties` per l'agent bridge Connect:Direct. Il trasferimento ha esito negativo e viene prodotto un messaggio di errore dopo che il numero di tentativi non riusciti ha raggiunto il valore di **cdMaxConnectionRetries property**. Per impostazione predefinita, il trasferimento viene tentato indefinitamente, con 60 secondi tra i tentativi.
- Se Managed File Transfer si è precedentemente collegato correttamente al nodo Connect:Direct come parte di questa richiesta di trasferimento, il trasferimento viene ritentato per un periodo di tempo determinato dai valori delle proprietà **cdMaxPartialWorkConnectionRetries** e **recoverableTransferRetryInterval**. Il trasferimento non riesce e viene prodotto un messaggio di errore, dopo che il numero di tentativi non riusciti raggiunge il valore della proprietà **cdMaxPartialWorkConnectionRetries**. Per impostazione predefinita, il trasferimento viene tentato indefinitamente, con 60 secondi tra i tentativi.
- Per alcuni tipi di errore del nodo Connect:Direct, ad esempio il nodo arrestato in modo forzato, i processi Connect:Direct passano allo stato `HEld Due to Error (HE)` quando il nodo viene ripristinato. Una volta ripristinato il nodo, Managed File Transfer riprende automaticamente tutti i processi Connect:Direct correlati al trasferimento file e con lo stato HE.
- Se il trasferimento ha esito negativo, tutti i file temporanei relativi al trasferimento vengono eliminati dal sistema che ospita il bridge Connect:Direct. L'ubicazione di questi file temporanei è definita dalla proprietà **cdTmpDir**.
- Se il trasferimento è da Managed File Transfer a Connect:Directe viene specificata una disposizione di origine di eliminazione, i file di origine non vengono eliminati se il trasferimento non riesce.

### Se le credenziali utente del nodo Connect:Direct non sono valide

Se Managed File Transfer non riesce a connettersi al nodo Connect:Direct perché le credenziali dell'utente vengono rifiutate dal nodo, il trasferimento non riesce e viene prodotto un messaggio di errore. In questa

situazione, verificare di aver fornito le credenziali utente corrette per il nodo Connect:Direct . Per ulteriori informazioni, consultare [Associazione delle credenziali per Connect:Direct](#).

## Se l'agent bridge Connect:Direct diventa non disponibile

Se l'agent bridge Connect:Direct diventa non disponibile, i trasferimenti file in corso vengono ripristinati allo stesso modo dei trasferimenti Managed File Transfer standard. Per ulteriori informazioni, consultare [“Ripristino e riavvio di MFT”](#) a pagina 300.

### Concetti correlati

[“Il bridge Connect:Direct”](#) a pagina 285

È possibile trasferire i file a e da una rete IBM Sterling Connect:Direct esistente. Utilizzare il bridge Connect:Direct , che è un componente di Managed File Transfer, per trasferire i file tra MFT e IBM Sterling Connect:Direct.

[“Ripristino e riavvio di MFT”](#) a pagina 300

Se l'agent o il gestore code non sono disponibili per qualsiasi motivo, ad esempio a causa di un errore di alimentazione o di rete, Managed File Transfer esegue il ripristino come riportato di seguito in questi scenari:

### Attività correlate

[Configurazione del bridge Connect:Direct](#)

### Riferimenti correlati

[Il file MFT agent.properties](#)

## Inoltro di un processo Connect:Direct definito dall'utente da una richiesta di trasferimento file

È possibile inviare una richiesta di trasferimento per un trasferimento che passa attraverso l'agent bridge Connect:Direct che richiama un processo Connect:Direct definito dall'utente come parte del trasferimento file.

Per impostazione predefinita, quando si inoltra una richiesta di trasferimento file per un trasferimento che passa attraverso il bridge Connect:Direct , l'agent bridge Connect:Direct genera il processo Connect:Direct utilizzato per trasferire il file al o dal nodo Connect:Direct remoto.

Tuttavia, è possibile configurare l'agent bridge Connect:Direct per richiamare un processo Connect:Direct definito dall'utente utilizzando il file `ConnectDirectProcessDefinition.xml` .

## Il file `ConnectDirectProcessDefinition.xml`

Il comando **fteCreateCDAgent** crea il file `ConnectDirectProcessDefinitions.xml` nella directory di configurazione dell'agent `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Prima di poter richiamare i processi Connect:Direct definiti dall'utente dall'agent bridge Connect:Direct , è necessario impostare definizioni di processi modificando questo file.

Il file definisce una o più serie di processi che includono l'ubicazione di uno o più processi Connect:Direct richiamati come parte di un trasferimento. Ogni serie di processi include un numero di condizioni. Se il trasferimento soddisfa tutte le condizioni della serie di processi, la serie di processi viene utilizzata per specificare quali processi Connect:Direct vengono richiamati dal trasferimento. Per ulteriori informazioni, consultare [“Specifiche del processo Connect:Direct da iniziare utilizzando il file `ConnectDirectProcessDefinition.xml`”](#) a pagina 295.

## Variabili simboliche intrinseche

È possibile utilizzare le variabili simboliche intrinseche definite da Managed File Transfer per sostituire i valori nei processi Connect:Direct definiti dall'utente. Per seguire la convenzione di denominazione Connect:Direct , tutte le variabili simboliche intrinseche utilizzate da Managed File Transfer hanno il formato `%FTE` seguito da cinque caratteri alfanumerici maiuscoli.

Quando si crea un processo per trasferire i file da un nodo Connect:Direct al sistema di bridge Connect:Direct, è necessario utilizzare la variabile intrinseca %FTETFILE come valore di TO FILE nel processo Connect:Direct. Quando si crea un processo per trasferire i file a un nodo Connect:Direct dal sistema di bridge Connect:Direct, è necessario utilizzare la variabile intrinseca %FTEFFILE come valore di FROM FILE nel processo Connect:Direct. Queste variabili contengono i percorsi di file temporanei che l'agent bridge Connect:Direct utilizza per i trasferimenti all'interno e all'esterno della rete Managed File Transfer.

Per ulteriori informazioni sulle variabili simboliche intrinseche, consultare la documentazione del prodotto Connect:Direct .

## Processi Connect:Direct di esempio

Managed File Transfer fornisce processi Connect:Direct di esempio. Questi esempi si trovano nella seguente directory: *MQ\_INSTALLATION\_PATH/mqft/samples/ConnectDirectProcessTemplates*.

### **Specifica del processo Connect:Direct da iniziare utilizzando il file *ConnectDirectProcessDefinition.xml***

Specificare quale processo Connect:Direct avviare come parte di un trasferimento Managed File Transfer . Managed File Transfer fornisce un file XML che è possibile modificare per specificare definizioni di processo.

## Informazioni su questa attività

Il comando **fteCreateCDAgent** crea il file *ConnectDirectProcessDefinitions.xml* nella directory di configurazione dell'agent *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/cd\_bridge\_agent\_name*. Prima di poter richiamare i processi Connect:Direct definiti dall'utente dall'agent bridge Connect:Direct , è necessario impostare definizioni di processi modificando questo file.

Per ciascun processo che si desidera specificare di richiamare come parte di un trasferimento tramite il bridge Connect:Direct , attenersi alla seguente procedura:

## Procedura

1. Definire il processo Connect:Direct che si desidera venga richiamato dall'agent bridge Connect:Direct come parte del trasferimento e salvare il template del processo in un file.
2. Aprire il file *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/cd\_bridge\_agent\_name/ConnectDirectProcessDefinitions.xml* in un editor di testo.
3. Creare un elemento `<processSet>` .
4. All'interno dell'elemento `<processSet>` , creare un elemento `<condition>` .
5. All'interno dell'elemento `<condition>` , creare uno o più elementi che definiscono una condizione che la richiesta di trasferimento deve corrispondere per richiamare il processo Connect:Direct definito nel passo 1. Questi elementi possono essere elementi `<match>` o `<defined>` .
  - Utilizzare un elemento `<match>` per specificare che il valore di una variabile deve corrispondere a un modello. Creare l'elemento `<match>` con i seguenti attributi:
    - `variable` - il nome della variabile di cui viene confrontato il valore. La variabile è un simbolo intrinseco. Per ulteriori informazioni, consultare [Variabili di sostituzione da utilizzare con processi Connect:Direct definiti dall'utente](#).
    - `value` - il modello da confrontare con il valore della variabile specificata.
    - Facoltativo: `pattern` - il tipo di pattern utilizzato dal valore dell'attributo `value` . Questo tipo di pattern può essere wildcard o regex. Questo attributo è facoltativo e il valore predefinito è `wildcard`.
  - Utilizzare un elemento `<defined>` per specificare che una variabile deve avere un valore definito. Creare l'elemento `<defined>` con il seguente attributo:

- *variable* - il nome della variabile che deve avere un valore definito. La variabile è un simbolo intrinseco. Per ulteriori informazioni, consultare [Variabili di sostituzione da utilizzare con processi Connect:Direct definiti dall'utente](#).

Le condizioni specificate nell'elemento <condition> sono combinate con un AND logico. Tutte le condizioni devono essere soddisfatte affinché l'agente bridge Connect:Direct richiami il processo specificato da questo elemento <processSet> . Se non si specifica un elemento <condition> , la serie di processi corrisponde a tutti i trasferimenti.

6. All'interno dell'elemento <processSet> , creare un elemento <process> .

7. All'interno dell'elemento <process> , creare un elemento <transfer> .

L'elemento transfer specifica il processo Connect:Direct che l'agent bridge Connect:Direct richiama come parte del trasferimento. Creare l'elemento <transfer> con il seguente attributo:

- *process* - - l'ubicazione del processo Connect:Direct definito nel passo 1. L'ubicazione di questo file è specificata con un percorso assoluto o relativo alla directory *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/cd\_bridge\_agent\_name* .

## Risultati

Quando si cerca una corrispondenza di condizione, l'agent bridge Connect:Direct esegue la ricerca dall'inizio del file alla fine del file. La prima corrispondenza trovata è quella utilizzata.

### Attività correlate

[Configurazione del bridge Connect:Direct](#)

### Riferimenti correlati

[Formato file di definizione del processo Connect:Direct](#)

[fteCreateCDAgent: crea un agent bridge Connect:Direct](#)

## ***Utilizzo di variabili simboliche intrinseche nei processi Connect:Direct richiamati da Managed File Transfer***

È possibile richiamare un processo Connect:Direct definito dall'utente da un trasferimento Managed File Transfer e trasmettere informazioni dal trasferimento al processo Connect:Direct utilizzando le variabili simboliche intrinseche nella definizione del processo.

### Informazioni su questa attività

Questo esempio utilizza variabili simboliche intrinseche per passare le informazioni da un trasferimento Managed File Transfer a un processo definito dall'utente Connect:Direct . Per ulteriori informazioni relative alle variabili simboliche intrinseche utilizzate da Managed File Transfer, consultare [Variabili di sostituzione da utilizzare con i processi Connect:Direct definiti dall'utente](#).

In questo esempio, il file viene trasferito da Managed File Transfer Agent a un nodo bridge Connect:Direct . La prima parte del trasferimento viene eseguita da Managed File Transfer. La seconda parte del trasferimento viene eseguita da un processo Connect:Direct definito dall'utente.

### Procedura

1. Creare un processo Connect:Direct che utilizza variabili simboliche intrinseche.

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. Salvare questo processo in un file di testo nella seguente ubicazione: *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/cd\_bridge\_agent/Example.cdp*
3. Modificare il file *ConnectDirectProcessDefinition.xml* in modo da includere una regola che richiama il processo *Connect:Direct* creato nel Passo 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="Example.cdp" />
    </tns:process>
  </tns:processSet>
</tns:cdprocess>
```

In questo esempio, se una richiesta di trasferimento viene inoltrata all'agent bridge *Connect:Direct* che ha *TOBERMORY* come nodo *Connect:Direct* di origine o di destinazione, viene richiamato il processo *Example.cdp* *Connect:Direct*.

4. Inoltrare una richiesta di trasferimento file che soddisfi le condizioni definite nel file *ConnectDirectProcessDefinition.xml* nel passo 3.  
Ad esempio,

```
fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                  -sm QM_WIMBLEDON -dm QM_COMMON
                  -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                  -sd leave c:\bungo\source.txt
```

In questo esempio, il nodo *Connect:Direct* di destinazione è *TOBERMORY*. Questo nodo è il nodo secondario nel trasferimento e il valore di *%FTESNODE* è impostato su *TOBERMORY*. Questo comando corrisponde alla condizione impostata nel file *ConnectDirectProcessDefinition.xml*.

5. *Managed File Transfer* trasferisce il file di origine in un'ubicazione temporanea sullo stesso sistema dell'agente bridge *Connect:Direct*.
6. L'agent bridge *Connect:Direct* imposta i valori delle variabili simboliche intrinseche dalle informazioni nella richiesta di trasferimento e nelle informazioni di configurazione.

Le variabili simboliche intrinseche sono impostati sui seguenti valori:

- *%FTEPNAME=nome\_processo* - Questo valore è un nome processo di 8 caratteri generato dall'agente bridge *Connect:Direct*.
- *%FTESNODE=TOBERMORY* - Questo valore è impostato dal parametro **-df** del comando **fteCreateTransfer**.
- *%FTEPUSER,=primary\_node\_user* - Queste informazioni vengono prese dal file *ConnectDirectCredentials.xml*.
- *%FTEPPASS=primary\_node\_user\_password* - Queste informazioni vengono prese dal file *ConnectDirectCredentials.xml*.
- *%FTESUSER,=secondary\_node\_user* - Queste informazioni vengono prese dal file *ConnectDirectCredentials.xml*.
- *%FTESPASS=secondary\_node\_user\_password* - Queste informazioni vengono prese dal file *ConnectDirectCredentials.xml*.
- *%FTEFFILE =ubicazione\_temporanea* - Questo valore è l'ubicazione temporanea del file sullo stesso sistema dell'agente bridge *Connect:Direct*.
- *%FTEFDISP=leave* - Questo valore è impostato dal parametro **-sd** del comando **fteCreateTransfer**.

- %FTETFILE=/home/bulgaria/destination.txt - Questo valore è impostato dal parametro **-df** del comando **fteCreateTransfer**.
  - %FTETDISP=overwrite - questo valore è impostato dal parametro **-de** del comando **fteCreateTransfer**.
7. Il processo Connect:Direct viene avviato sul nodo bridge Connect:Direct. Connect:Direct trasferisce il file dall'ubicazione temporanea sul sistema bridge Connect:Direct alla destinazione /home/bulgaria/destination.txt sul sistema su cui è in esecuzione il nodo Connect:Direct TOBERMORY.

### Concetti correlati

[“Inoltro di un processo Connect:Direct definito dall'utente da una richiesta di trasferimento file” a pagina 294](#)

È possibile inviare una richiesta di trasferimento per un trasferimento che passa attraverso l'agent bridge Connect:Direct che richiama un processo Connect:Direct definito dall'utente come parte del trasferimento file.

### Riferimenti correlati

[Variabili di sostituzione da utilizzare con processi Connect:Direct definiti dall'utente](#)

## Utilizzo dei processi Connect:Direct per inoltrare richieste di trasferimento Managed File Transfer

È possibile inviare una richiesta di trasferimento all'agent bridge Connect:Direct da un processo Connect:Direct. Managed File Transfer fornisce comandi che possono essere richiamati da un'istruzione **RUN TASK** in un processo Connect:Direct.

Managed File Transfer fornisce i seguenti comandi da utilizzare con i processi Connect:Direct:

### ftetag

Specificare questo comando in un passo che precede il comando **ftebxf** o **ftexcfer** per creare le informazioni di verifica richieste per il trasferimento. Questo comando utilizza la specifica di origine del trasferimento come parametro. Per informazioni sul formato della specifica di origine, consultare [fteCreateTransfer: avvio di un nuovo trasferimento file](#).

### ftebxf

Specificare questo comando per creare una richiesta di trasferimento file quando il gestore code a cui viene inoltrata la richiesta di trasferimento si trova sullo stesso sistema del nodo Connect:Direct che inoltra il comando. Questo comando utilizza gli stessi parametri del comando **fteCreateTransfer**. Per informazioni su questi parametri, vedere [fteCreateTransfer: avviare un nuovo trasferimento file](#). Questo comando ha anche un ulteriore parametro:

#### **-qmgrname**

Obbligatorio. Il nome del gestore code a cui inoltrare il comando.

### ftexcfer

Specificare questo comando per creare una richiesta di trasferimento file quando il gestore code a cui viene inviata la richiesta di trasferimento si trova su un sistema diverso dal nodo Connect:Direct che inoltra il comando. Questo comando utilizza gli stessi parametri del comando **fteCreateTransfer**. Per informazioni sui parametri, consultare [fteCreateTransfer: avviare un nuovo trasferimento file](#). Questo comando ha anche tre parametri aggiuntivi:

#### **-qmgrname**

Obbligatorio. Il nome del gestore code a cui inoltrare il comando.

#### **-nomeconnessione**

Obbligatorio. L'host e la porta del gestore code a cui inoltrare il comando, specificati in formato CONNAME IBM MQ. Ad esempio, host.example.com(1337).

#### **-nomecanale**

Facoltativo. Il nome del canale da utilizzare per connettersi al gestore code a cui inoltrare il comando. Se non viene specificato, viene utilizzato il valore predefinito SYSTEM.DEF.SVRCONN.

## Attività correlate

“Creazione e inoltro di un processo Connect:Direct che richiama Managed File Transfer utilizzando il richiedente Connect:Direct.” a pagina 299

Il richiedente Connect:Direct è una GUI (graphical user interface) che è possibile utilizzare per creare e inoltrare un processo Connect:Direct che richiama Managed File Transfer.

## Riferimenti correlati

Esempio: un file di elaborazione Connect:Direct che richiama i comandi MFT

## **Creazione e inoltro di un processo Connect:Direct che richiama Managed File Transfer utilizzando il richiedente Connect:Direct .**

Il richiedente Connect:Direct è una GUI (graphical user interface) che è possibile utilizzare per creare e inoltrare un processo Connect:Direct che richiama Managed File Transfer.

## Informazioni su questa attività

Questa attività descrive come creare un processo Connect:Direct che richiama il comando Managed File Transfer **ftecxfer** o il comando **ftebxfer** . Utilizzare il comando **ftecxfer** quando il gestore code a cui è inoltrata la richiesta di trasferimento si trova su un sistema differente rispetto al nodo Connect:Direct che inoltra il comando. Utilizzare il comando **ftebxfer** quando il gestore code a cui viene inoltrata la richiesta di trasferimento si trova sullo stesso sistema del nodo Connect:Direct che inoltra il comando. Il comando **ftecxfer** effettua una connessione client al gestore code agent dell'agent di origine del trasferimento. Prima di richiamare il comando **ftecxfer** , è necessario richiamare il comando **ftetag** e trasmettergli le informazioni sulla specifica di origine. Ciò consente al processo di essere registrato e controllato allo stesso modo dei trasferimenti avviati da Managed File Transfer.

## Procedura

1. Avviare il richiedente Connect:Direct .
2. Nella scheda **Nodi** del pannello, selezionare il nodo Connect:Direct utilizzato come nodo primario del processo.
3. Selezionare **File > Nuovo > Processo**. Viene visualizzata la finestra **Proprietà del processo** .
4. Nel campo **Nome:** , immettere il nome del processo.
5. Selezionare il nodo secondario dall'elenco **Snodo > Nome** .
6. Scegliere il sistema operativo del nodo secondario dall'elenco **Snodo > Sistema operativo** .
7. Opzionale: Completare tutte le ulteriori informazioni richieste in questa finestra.
8. Fare clic su **OK**. La finestra **Proprietà processo** viene chiusa.
9. Creare un'istruzione che esegue il comando Managed File Transfer **ftetag** .
  - a) Fare clic con il tasto destro del mouse nella finestra **Processo** sull'istruzione **End** .
  - b) Selezionare **Inserisci > Esegui attività**. Viene visualizzata la finestra **Esegui istruzione attività** .
  - c) Nel campo **Etichetta:** , immettere Tag.
  - d) Nel campo **Parametri o comandi facoltativi** , immettere `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)`. Per ulteriori informazioni sul formato di `source_specification`, consultare **ftecCreateTransfer: start a new file transfer**.
  - e) Fare clic su **OK**. La finestra **Esegui istruzione attività** viene chiusa.
10. Creare un'istruzione che esegue il comando Managed File Transfer **ftecxfer** o **ftebxfer** .
  - a) Fare clic con il tasto destro del mouse nella finestra **Processo** sull'istruzione **End** .
  - b) Selezionare **Inserisci > Esegui attività**. Viene visualizzata la finestra **Esegui istruzione attività** .
  - c) Nel campo **Etichetta:** , immettere Transfer.
  - d) Nel campo **Parametri o comandi facoltativi** , immettere `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` o `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` in base al comando scelto. I parametri utilizzati dai comandi **ftecxfer**

e **ftebxfer** sono gli stessi utilizzati dal comando **fteCreateTransfer** , oltre ad alcuni parametri aggiuntivi specifici di **ftecxfer** e **ftebxfer**. Per ulteriori informazioni, consultare **fteCreateTransfer**: avviare un nuovo trasferimento file e [“Utilizzo dei processi Connect:Direct per inoltrare richieste di trasferimento Managed File Transfer”](#) a pagina 298.

- e) Fare clic su **OK**. La finestra **Esegui istruzione attività** viene chiusa.
11. Opzionale: Creare eventuali istruzioni aggiuntive richieste.
12. Inoltrare il processo.
  - a) Fare clic con il tasto destro del mouse nella finestra **Processo** .
  - b) Selezionare **Inoltra**. Viene visualizzata la finestra **Connect:Direct Allega** .
  - c) Immettere il nome utente e la password da utilizzare per eseguire il processo.
  - d) Fare clic su **OK**.

### Concetti correlati

[“Utilizzo dei processi Connect:Direct per inoltrare richieste di trasferimento Managed File Transfer”](#) a pagina 298

È possibile inviare una richiesta di trasferimento all'agent bridge Connect:Direct da un processo Connect:Direct . Managed File Transfer fornisce comandi che possono essere richiamati da un'istruzione **RUN TASK** in un processo Connect:Direct .

## Utilizzo di MFT da IBM Integration Bus

È possibile utilizzare Managed File Transfer da IBM Integration Bus utilizzando i nodi FTEOutput e FTEInput.

- Utilizzare il nodo FTEInput per trasferire un file sulla rete utilizzando Managed File Transfer ed elaborare tale file come parte di un flusso Integration Bus .
- Utilizzare il nodo FTEOutput per trasferire un file che è stato emesso da un flusso Integration Bus in un'altra ubicazione nella rete.

Gli agent che trasferiscono i file da o verso l'agent del broker possono essere a qualsiasi livello di Managed File Transfer.

Per ulteriori informazioni, fare riferimento alla [documentazione del prodotto IBM Integration Bus](#).

## Ripristino e riavvio di MFT

Se l'agent o il gestore code non sono disponibili per qualsiasi motivo, ad esempio a causa di un errore di alimentazione o di rete, Managed File Transfer esegue il ripristino come riportato di seguito in questi scenari:

- In genere, se si verifica un problema durante il trasferimento di un file, Managed File Transfer recupera e riavvia il trasferimento file dopo che il problema è stato risolto.
- Se un file che era in fase di trasferimento viene eliminato o modificato mentre l'agent o il gestore code non sono disponibili, il trasferimento ha esito negativo e si riceve un messaggio nel log di trasferimento che fornisce i dettagli sull'errore.
- Se un processo agent non riesce durante un trasferimento file, il trasferimento continua quando si riavvia l'agent.
- Se un agent perde la connessione al gestore code dell'agent, l'agent attende durante il tentativo di riconnettersi al gestore code. Quando l'agent si riconnette correttamente al gestore code, il trasferimento corrente continua.
- Se l'agent viene arrestato per qualsiasi motivo, tutti i monitoraggi delle risorse associati ad un agent arrestano il polling. Quando l'agent viene ripristinato, vengono riavviati anche i monitoraggi e riprende il polling delle risorse.
- Per un trasferimento file con una disposizione di origine delete, se si verifica un ripristino dopo che tutti i dati vengono inviati da un agent di origine a un agent di destinazione, il file di origine viene sbloccato prima dell'eliminazione. Questo sblocco indica che il file di origine potrebbe essere modificato

prima che il file venga eliminato. Pertanto, è considerato non sicuro eliminare il file di origine e viene visualizzata la seguente avvertenza:

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

In questo caso, verificare che il contenuto del file di origine non sia stato modificato e quindi eliminare manualmente il file di origine.

Puoi verificare lo stato dei tuoi trasferimenti in IBM MQ Explorer. Se i trasferimenti vengono visualizzati come `Stalled`, potrebbe essere necessario intraprendere un'azione correttiva perché lo stato di stallo denota un problema con l'agent o tra i due agent coinvolti nel trasferimento.

### Attività correlate

[“Impostazione di una scadenza per il recupero di trasferimenti in stallo” a pagina 301](#)

È possibile impostare un timeout di ripristino del trasferimento per i trasferimenti di file in stallo che si applica a tutti i trasferimenti per un agent di origine. È anche possibile impostare un timeout di ripristino trasferimento per un singolo trasferimento. Se si imposta un periodo di tempo specifico, in secondi, durante il quale un agent di origine tenta di ripristinare un trasferimento file in stallo e il trasferimento non riesce quando l'agent raggiunge il timeout, il trasferimento non riesce.

## Impostazione di una scadenza per il recupero di trasferimenti in stallo

È possibile impostare un timeout di ripristino del trasferimento per i trasferimenti di file in stallo che si applica a tutti i trasferimenti per un agent di origine. È anche possibile impostare un timeout di ripristino trasferimento per un singolo trasferimento. Se si imposta un periodo di tempo specifico, in secondi, durante il quale un agent di origine tenta di ripristinare un trasferimento file in stallo e il trasferimento non riesce quando l'agent raggiunge il timeout, il trasferimento non riesce.

### Informazioni su questa attività

Da IBM MQ 9.1, è possibile impostare un timeout di ripristino trasferimento che si applichi a tutti i trasferimenti per un agent di origine aggiungendo un parametro di timeout di ripristino trasferimento al file `agent.properties` dell'agent. È anche possibile impostare un timeout di ripristino del trasferimento per un singolo trasferimento dalla riga comandi o con IBM MQ Explorer oppure utilizzando le attività Apache Ant. Se è impostato un valore di timeout di ripristino trasferimento nel file `agent.properties`, l'impostazione del timeout di ripristino trasferimento per un singolo trasferimento sovrascrive il valore nel file `agent.properties`.

Esistono tre opzioni per il timeout del ripristino del trasferimento:

- L'agent continua a tentare di recuperare il trasferimento bloccato fino a quando non viene completato correttamente. È uguale al comportamento predefinito dell'agent se il timeout di ripristino del trasferimento non è impostato.
- L'agent contrassegna il trasferimento come non riuscito immediatamente dopo l'avvio del ripristino.
- L'agent continua a ripetere il trasferimento in stallo per un periodo di tempo specificato prima che il trasferimento venga contrassegnato come non riuscito.

L'impostazione del timeout di ripristino del trasferimento file è facoltativa. Se non viene impostato, i trasferimenti seguono il comportamento predefinito. Questo è lo stesso comportamento predefinito di un Managed File Transfer agent di origine prima di IBM MQ 9.1, in cui l'agent continua a tentare di ripristinare un trasferimento bloccato fino a quando non viene eseguito correttamente.

### Concetti correlati

[“Ripristino e riavvio di MFT” a pagina 300](#)

Se l'agent o il gestore code non sono disponibili per qualsiasi motivo, ad esempio a causa di un errore di alimentazione o di rete, Managed File Transfer esegue il ripristino come riportato di seguito in questi scenari:

## Concetti di timeout di ripristino trasferimento

È possibile impostare la quantità di tempo, in secondi, durante la quale un agent di origine continua a tentare di ripristinare un trasferimento file bloccato. Se il trasferimento non ha esito positivo quando l'agent raggiunge il timeout per l'intervallo di tentativi, il trasferimento ha esito negativo.

### Precedenza timeout di recupero

Un valore di timeout del ripristino del trasferimento per un singolo trasferimento specificato tramite i comandi **fteCreateTransfer**, **fteCreateTemplate** o **fteCreateMonitor** oppure utilizzando IBM MQ Explorer specificato nell'elemento nidificato **fte:filespec**, ha la precedenza sul valore specificato per il parametro **transferRecoveryTimeout** nel file `agent.properties` per l'agent di origine.

Ad esempio, se il comando **fteCreateTransfer** viene avviato senza la coppia parametro e valore **-rt**, l'agent di origine AGENT1 ricerca nel file `agent.properties` un valore **transferRecoveryTimeout** per determinare il funzionamento del timeout di ripristino:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

Se il parametro **transferRecoveryTimeout** nel file `agent.properties` non è impostato o è impostato su **-1**, l'agent segue il comportamento predefinito e tenta di ripristinare il trasferimento fino a quando non viene eseguito correttamente.

Tuttavia, se il comando **fteCreateTransfer** include il parametro **-rt**, il valore di questo parametro ha la precedenza sul valore nel file `agent.properties` e viene utilizzato come impostazione di timeout di ripristino per il trasferimento:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

### Contatore timeout di ripristino

Il contatore del timeout di ripristino viene avviato quando il trasferimento entra nello stato di recupero. Viene pubblicato un messaggio di log di trasferimento nel SISTEMA SYSTEM.FTE con la stringa di argomenti `Log/agent_name/transfer_ID` per indicare che lo stato del trasferimento è stato modificato in ripristino e l'ora dell'orologio dell'agent di origine in cui è stato modificato lo stato. Se il trasferimento viene ripreso entro l'intervallo di tentativi impostato e non raggiunge il timeout di recupero (contatore  $\leq$  timeout di ripristino), il contatore viene reimpostato su 0, pronto per iniziare di nuovo se il trasferimento entra nel recupero.

Se il contatore raggiunge il valore massimo impostato per il timeout di ripristino (contatore  $==$  timeout di ripristino), il recupero del trasferimento si arresta e l'agent di origine riporta il trasferimento come non riuscito. Questo tipo di errore di trasferimento, causato dal fatto che il trasferimento ha raggiunto il timeout di recupero, è indicato dal codice messaggio, `RECOVERY TIMEOUT (69)`. Un altro messaggio del log di trasferimento viene pubblicato nel SISTEMA SYSTEM.FTE FTE, con una stringa di argomento `Log/agent_name/transfer_ID`, per indicare che il trasferimento non è riuscito e include un messaggio, il codice di ritorno e il log eventi dell'agent di origine. Il log degli eventi dell'agente di origine viene aggiornato con un messaggio quando si verifica uno dei seguenti eventi durante il recupero:

- Quando il parametro di timeout del ripristino è impostato su un valore maggiore di **-1**, il trasferimento avvia il ripristino. Il log eventi dell'agent viene aggiornato per indicare l'inizio del timer di recupero per **TransferId** e la quantità di tempo in cui l'agent di origine attende prima di avviare l'elaborazione del timeout di recupero.
- Quando il trasferimento di recupero viene ripreso, il log eventi dell'agent di origine viene aggiornato con un nuovo messaggio per indicare che il **TransferId** che era in ripristino viene ripreso.
- Quando un trasferimento di ripristino è scaduto, il log eventi dell'agent di origine viene aggiornato per indicare il **TransferId** che non è riuscito durante il recupero a causa di un timeout di recupero.

Questi messaggi di log consentono agli utenti (sottoscrittori e logger) di identificare i trasferimenti non riusciti a causa del supero tempo di recupero del trasferimento.

Il contatore per il timeout di ripristino si trova sempre sull'agent di origine. Tuttavia, se l'agent di destinazione non riesce a ricevere informazioni dall'agent di origine in modo tempestivo, può inviare una richiesta all'agent di origine per inserire il trasferimento nel ripristino. Per un trasferimento in cui è impostata l'opzione di timeout di ripristino, l'agent di origine avvia il contatore di timeout di recupero quando riceve la richiesta dall'agent di destinazione.

La gestione manuale è ancora richiesta per i trasferimenti che non utilizzano l'opzione di timeout di ripristino, i trasferimenti non riusciti e parzialmente completi.

Per le serie di trasferimento, dove una singola richiesta di trasferimento viene emessa per più file e alcuni dei file sono stati completati correttamente ma solo parzialmente, il trasferimento è ancora contrassegnato come non riuscito poiché non è stato completato come previsto. L'agent di origine potrebbe essere scaduto durante il trasferimento del file parzialmente completato.

Verificare che l'agent di destinazione e il server file siano pronti e in uno stato per accettare i trasferimenti file.

È necessario emettere nuovamente la richiesta di trasferimento per l'intera serie, ma per evitare problemi poiché alcuni dei file rimangono sulla destinazione dal tentativo di trasferimento iniziale, è possibile emettere la nuova richiesta con l'opzione `sovrascrivi se esistente` specificata. Ciò garantisce che la serie incompleta di file dal tentativo di trasferimento precedente venga ripulita come parte del nuovo trasferimento, prima che i file vengano scritti nuovamente nella destinazione.

**V 9.2.0** Da IBM MQ 9.1.5, non è più necessario rimuovere manualmente i file parte lasciati su una destinazione dopo un tentativo di trasferimento iniziale non riuscito. Se un timeout di ripristino del trasferimento è impostato per un trasferimento, l'agent di origine sposta il trasferimento nello stato `RecoveryTimedOut` se il ripristino del trasferimento va in timeout. Dopo che il trasferimento è stato risincronizzato, l'agent di destinazione rimuove tutti i file parte creati durante il trasferimento e invia un messaggio di completamento all'agent di origine.

## Tracce e messaggi

I punti di traccia sono inclusi per scopi diagnostici. Vengono registrati il valore di timeout di ripristino, l'inizio dell'intervallo di nuovi tentativi, l'avvio del periodo di ripresa e la reimpostazione del contatore e se il trasferimento è andato in timeout e non è riuscito. In caso di un problema o di un comportamento imprevisto, è possibile raccogliere i file di traccia e di log di output dell'agent di origine e fornirli quando richiesto dal supporto IBM, per facilitare la risoluzione dei problemi.

I messaggi notificano quando:

- Un trasferimento avvia il ripristino (`BFGTR0081I`)
- Un trasferimento è terminato perché si è verificato un timeout dal ripristino (`BFGSS0081E`)
- L'operazione di trasferimento riprende dopo essere stata ripristinata (`BFGTR0082I`)

### Concetti correlati

[“Ripristino e riavvio di MFT” a pagina 300](#)

Se l'agent o il gestore code non sono disponibili per qualsiasi motivo, ad esempio a causa di un errore di alimentazione o di rete, Managed File Transfer esegue il ripristino come riportato di seguito in questi scenari:

## Impostazione del timeout di ripristino del trasferimento per tutti i trasferimenti per un agent di origine

È possibile impostare un timeout di ripristino del trasferimento che si applica a tutti i trasferimenti per un agent di origine aggiungendo il parametro `transferRecoveryTimeout` al file `agent.properties`.

## Informazioni su questa attività

Per impostare un timeout di ripristino transfer che si applica a tutti i trasferimenti per un agent di origine, aggiungere la coppia parametro e valore per **transferRecoveryTimeout** al file `agent.properties`.

Esistono tre opzioni per il parametro **transferRecoveryTimeout** :

**-1**

L'agent continua a tentare di recuperare il trasferimento bloccato fino a quando il trasferimento non viene eseguito correttamente. L'uso di questa opzione equivale al comportamento predefinito dell'agente quando la proprietà non è impostata.

**0**

L'agent arresta il trasferimento file non appena avvia il ripristino.

**>0**

L'agent continua a tentare di recuperare il trasferimento bloccato per il periodo di tempo in secondi come impostato dal valore intero positivo specificato.

Tutte le modifiche apportate al file `agent.properties` diventano effettive solo dopo il riavvio dell'agent.

Se richiesto, è possibile sovrascrivere il valore di timeout del ripristino del trasferimento nel file `agent.properties` per un singolo trasferimento. Per ulteriori informazioni, consultare [“Impostazione del timeout di ripristino del trasferimento per singoli trasferimenti”](#) a pagina 304.

## Procedura

- Per specificare che l'agent continua a tentare di ripristinare il trasferimento in stallo fino a quando non viene completato correttamente, impostare un valore di timeout di ripristino del trasferimento **-1** come mostrato nel seguente esempio:

```
transferRecoveryTimeout=-1
```

- Per specificare che l'agent contrassegna il trasferimento come non riuscito immediatamente dopo l'immissione del ripristino, impostare il valore di timeout del ripristino del trasferimento **0** come mostrato nel seguente esempio:

```
transferRecoveryTimeout=0
```

- Per specificare che l'agent continua a ritentare un trasferimento bloccato per un determinato periodo di tempo prima che il trasferimento venga contrassegnato come non riuscito, impostare un valore di timeout di ripristino del trasferimento per il periodo di tempo, in secondi, in cui si desidera che l'agent continui a ritentare.

Ad esempio, l'impostazione di un valore di timeout di ripristino del trasferimento **21600** significa che l'agent continua a tentare di ripristinare il trasferimento per sei ore da quando avvia il ripristino:

```
transferRecoveryTimeout=21600
```

Il valore massimo per questo parametro è 999999999.

## Impostazione del timeout di ripristino del trasferimento per singoli trasferimenti

È possibile impostare il timeout di ripristino trasferimento per un singolo trasferimento dalla riga comandi o con IBM MQ Explorer oppure utilizzando le attività Apache Ant. Se è impostato un valore di timeout di ripristino trasferimento nel file `agent.properties`, l'impostazione del timeout di ripristino trasferimento per un singolo trasferimento sovrascrive il valore impostato nel file `agent.properties`.

## Informazioni su questa attività

È possibile impostare il parametro di timeout del ripristino del trasferimento per un singolo trasferimento quando si è:

- Creazione di un trasferimento utilizzando il comando **fteCreateTransfer** o utilizzando IBM MQ Explorer.
- Creazione di un modello di trasferimento utilizzando il comando **fteCreateTemplate** o utilizzando IBM MQ Explorer.
- Creazione di un monitoraggio delle risorse utilizzando il comando **fteCreateMonitor** o utilizzando IBM MQ Explorer.
- Copia o spostamento dei file utilizzando le attività `fte: filecopy` o `fte: filemove` Ant .

Se si imposta un valore di timeout del ripristino del trasferimento per un singolo trasferimento, questo valore sovrascrive il valore di timeout del ripristino del trasferimento impostato nel file `agent.properties` (consultare ["Impostazione del timeout di ripristino del trasferimento per tutti i trasferimenti per un agente di origine"](#) a pagina 303).

## Procedura

- Per utilizzare il comando **fteCreateTransfer** o **fteCreateTemplate** per impostare il timeout di ripristino del trasferimento, specificare l'opzione appropriata per il parametro **-rt** :
  - 1**  
L'agent continua a tentare di recuperare il trasferimento bloccato fino a quando il trasferimento non viene eseguito correttamente. L'uso di questa opzione equivale al comportamento predefinito dell'agente quando la proprietà non è impostata.
  - 0**  
L'agent arresta il trasferimento file non appena avvia il ripristino.
  - >0**  
L'agent continua a tentare di ripristinare il trasferimento in stallo per la quantità di tempo specificata in secondi.

### Esempi per il comando **fteCreateTransfer**

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 0 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

### Esempi per il comando **fteCreateTemplate**

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt -1 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 21600 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

Non è disponibile alcun parametro **-rt** per il comando **fteCreateMonitor** . Se si imposta il parametro **-rt** con il comando **fteCreateTransfer** e si imposta anche il parametro **-gt** , il parametro di timeout del recupero viene incluso nel documento XML con la definizione di trasferimento generata quando si esegue il comando **fteCreateTransfer** . Il monitoraggio risorse

utilizza questo documento XML quando si esegue il comando **fteCreateMonitor** . Nell'esempio seguente, i dettagli del timeout di ripristino del trasferimento vengono inclusi nel file `task.xml` :

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr "fileSize>=5MB,*.zip"
```

- Per utilizzare la pagina della procedura guidata IBM MQ Explorer Nuovo trasferimento, Nuovo monitoraggio o Nuovo template per impostare il timeout di ripristino del trasferimento, selezionare l'opzione richiesta nel campo **Timeout ripristino trasferimento** (secondi):

#### Come agent di origine

Se si seleziona **Come agent di origine**, il valore del parametro **transferRecoveryTimeout** dal file `agent.properties` viene utilizzato se è impostato, altrimenti viene applicato il comportamento predefinito per il timeout di ripristino del trasferimento.

#### Casella di elenco numerica

Se si immette un tempo in secondi nella casella di elenco numerica, l'agent continua a tentare di ripristinare il trasferimento bloccato per il periodo di tempo specificato.

#### Nessuno

Se si seleziona **Nessuno**, non viene impostato alcun timeout di ripristino del trasferimento e l'agent continua a tentare di ripristinare il trasferimento bloccato fino a quando il trasferimento non viene eseguito correttamente.

- Per impostare il timeout di ripristino utilizzando le attività Ant . includere l'opzione e il valore **transferRecoveryTimeout** , con gli elementi **fte:filecopy** o **fte:filemove** per spostare o copiare i file, ad esempio:

#### Esempio per **fte:filecopy**

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result" transferRecoveryTimeout="0">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

#### Esempio per **fte:filemove**

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="move.result" transferRecoveryTimeout="21600">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

Linux

Windows

AIX

## AmministrazioneMQ Telemetry

MQ Telemetry viene gestito utilizzando IBM MQ Explorer o su una riga comandi. Utilizzare Esplora per configurare i canali di telemetria, controllare il servizio di telemetria e monitorare i client MQTT connessi a IBM MQ. Configurare la sicurezza di MQ Telemetry utilizzando JAAS, TLS e il gestore di autorizzazione oggetto IBM MQ .

### Amministrazione mediante IBM MQ Explorer

Utilizzare Esplora per configurare i canali di telemetria, controllare il servizio di telemetria e monitorare i client MQTT connessi a IBM MQ. Configurare la sicurezza di MQ Telemetry utilizzando JAAS, TLS e il gestore di autorizzazione oggetto IBM MQ .

### Gestione mediante la riga comandi

MQ Telemetry può essere completamente gestito dalla riga comandi utilizzando i comandi MQSC.

La documentazione MQ Telemetry contiene anche script di esempio che dimostrano l'utilizzo di base dell'applicazione client IBM MQ Telemetry Transport v3 .

Prima di utilizzarli, leggere e comprendere gli esempi in [IBM MQ Telemetry Transport programmi di esempio](#) .

### Concetti correlati

[MQ Telemetry](#)

### Riferimenti correlati

[Proprietà MQXR](#)

Linux

AIX

## Configurazione di un gestore code per la telemetria su

### Linux e AIX

Seguire questa procedura per configurare MQ Telemetry manualmente. Se è necessaria solo una configurazione semplice che utilizza l'ID utente guest, è possibile eseguire la procedura guidata di supporto MQ Telemetry in IBM MQ Explorer.

### Prima di iniziare

Se hai bisogno solo di una semplice configurazione, considera l'utilizzo del supporto MQ Telemetry in IBM MQ Explorer. Questo supporto include una procedura guidata e una procedura di comando di esempio `sampleMQM`. Queste risorse impostano una configurazione iniziale utilizzando l'ID utente guest. Consultare [Verifica dell'installazione di MQ Telemetry utilizzando programmi di esempio IBM MQ Explorer e IBM MQ Telemetry Transport](#).

Se è necessaria una configurazione più complessa che utilizza un metodo di autenticazione diverso, utilizzare i passi in questa attività. Iniziare con i seguenti passi iniziali:

1. Per informazioni su come installare IBM MQ e la funzione MQ Telemetry , consultare [Considerazioni sull'installazione per MQ Telemetry](#) .
2. Creare e avviare un gestore code. Il gestore code viene indicato come `qMgr` in questa attività.
3. Come parte di questa attività, configurare il servizio di telemetria (MQXR). Le impostazioni delle proprietà MQXR sono memorizzate in un file delle proprietà specifico della piattaforma: `mqxr_win.properties`. Normalmente non è necessario modificare direttamente il file delle proprietà MQXR, poiché quasi tutte le impostazioni possono essere configurate tramite i comandi di gestione MQSC o IBM MQ Explorer. Se si decide di modificare direttamente il file, arrestare il gestore code prima di apportare le modifiche. Vedere [proprietà MQXR](#).

### Informazioni su questa attività

Seguire i passi in questa attività per configurare manualmente MQ Telemetry utilizzando diversi schemi di autorizzazione.

### Procedura

1. Aprire una finestra comandi nella directory degli esempi di telemetria.

La directory degli esempi di telemetria è `/opt/mqm/mqxr/samples`.

2. Creare la coda di trasmissione telemetria.

Se `SYSTEM.MQTT.TRANSMIT.QUEUE` non esiste, viene creato automaticamente quando il servizio di telemetria (MQXR) viene avviato per la prima volta e impostato per utilizzare l'ID utente guest. Tuttavia, questa attività configura MQ Telemetry per utilizzare uno schema di autorizzazione differente. Per questa attività, creare `SYSTEM.MQTT.TRANSMIT.QUEUE` e configurare l'accesso ad esso, prima di avviare il servizio di telemetria (MQXR).

Esegui il seguente comando:

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

### 3. Impostare la coda di trasmissione predefinita.

È più semplice inviare messaggi direttamente ai client MQTT se `SYSTEM.MQTT.TRANSMIT.QUEUE` è la coda di trasmissione predefinita. Altrimenti, è necessario aggiungere una definizione di coda remota per ogni client che riceve messaggi IBM MQ; consultare [“Invio diretto di un messaggio ad un client” a pagina 312](#). Notare che la modifica della coda di trasmissione predefinita potrebbe interferire con la configurazione esistente.

Quando il servizio di telemetria (MQXR) viene avviato per la prima volta, non imposta `SYSTEM.MQTT.TRANSMIT.QUEUE` come coda di trasmissione predefinita per il gestore code. Per configurare questa impostazione, modificare la proprietà della coda di trasmissione predefinita. A tale scopo, utilizzare IBM MQ Explorer o eseguire il seguente comando:

```
echo "ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') " | runmqsc qMgr
```

### 4. Seguire una procedura in [“Autorizzazione dei client MQTT ad accedere agli oggetti di IBM MQ” a pagina 314](#) per creare uno o più ID utente. Gli ID utente hanno l'autorità di pubblicare, sottoscrivere e inviare pubblicazioni ai client MQTT.

### 5. Installare il servizio di telemetria (MQXR).

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc qMgr
```

Vedere anche il codice di esempio in [“Creazione del SYSTEM.MQXR.SERVICE” a pagina 308](#).

### 6. Avviare il servizio.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE)" | runmqsc qMgr
```

Il servizio MQXR (telemetria) viene avviato automaticamente all'avvio del gestore code. Viene avviato manualmente in questa attività, poiché il gestore code è già in esecuzione.

### 7. Utilizzando IBM MQ Explorer, configurare i canali di telemetria per accettare le connessioni dai client MQTT.

I canali di telemetria devono essere configurati in modo che le loro identità siano uno degli ID utente definiti nel passo [“4” a pagina 308](#).

Vedere anche `DEFINE CHANNEL (MQTT)`.

### 8. Verificare la configurazione eseguendo il client di esempio.

Per consentire al client di esempio di utilizzare il canale di telemetria, il canale deve autorizzare il client a pubblicare, sottoscrivere e ricevere pubblicazioni. Il client di esempio si connette al canale di telemetria sulla porta 1883 per impostazione predefinita. Vedere inoltre [IBM MQ Telemetry Transport programmi di esempio](#).

## Creazione del SYSTEM.MQXR.SERVICE

Utilizzare il comando `runMQXRService` per creare il `SYSTEM.MQXR.SERVICE`.

LTS

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

V9.2.4

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
```

```
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+" -sf "[DEFAULT]"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

**Nota:** **V 9.2.4** Da IBM MQ 9.2.4 in poi, l'indicatore **-sf** fornisce il nome file della chiave delle credenziali che contiene la chiave da utilizzare per crittografare le passphrase del canale TLS. Per ulteriori informazioni, vedi [Encryption of passphrase for MQTT TLS channels](#).

## Windows Configurazione di un gestore code per la telemetria su Windows

Seguire questa procedura per configurare MQ Telemetry manualmente. Se è necessaria solo una configurazione semplice che utilizza l'ID utente guest, è possibile eseguire la procedura guidata di supporto MQ Telemetry in IBM MQ Explorer.

### Prima di iniziare

Se hai bisogno solo di una semplice configurazione, considera l'utilizzo del supporto MQ Telemetry in IBM MQ Explorer. Questo supporto include una procedura guidata e una procedura di comando di esempio `sampleMQM`. Queste risorse impostano una configurazione iniziale utilizzando l'ID utente guest. Consultare [Verifica dell'installazione di MQ Telemetry utilizzando programmi di esempio IBM MQ Explorer e IBM MQ Telemetry Transport](#).

Se è necessaria una configurazione più complessa che utilizza un metodo di autenticazione diverso, utilizzare i passi in questa attività. Iniziare con i seguenti passi iniziali:

1. Per informazioni su come installare IBM MQ e la funzione MQ Telemetry, consultare [Considerazioni sull'installazione per MQ Telemetry](#).
2. Creare e avviare un gestore code. Il gestore code viene indicato come *qMgr* in questa attività.
3. Come parte di questa attività, configurare il servizio di telemetria (MQXR). Le impostazioni delle proprietà MQXR sono memorizzate in un file delle proprietà specifico della piattaforma: `mqxr_win.properties`. Normalmente non è necessario modificare direttamente il file delle proprietà MQXR, poiché quasi tutte le impostazioni possono essere configurate tramite i comandi di gestione MQSC o IBM MQ Explorer. Se si decide di modificare direttamente il file, arrestare il gestore code prima di apportare le modifiche. Vedere [proprietà MQXR](#).

### Informazioni su questa attività

Seguire i passi in questa attività per configurare manualmente MQ Telemetry utilizzando diversi schemi di autorizzazione.

### Procedura

1. Aprire una finestra comandi nella directory degli esempi di telemetria.

La directory degli esempi di telemetria è *WMQ program installation directory\mqxr\samples*.

2. Creare la coda di trasmissione telemetria.

Se `SYSTEM.MQTT.TRANSMIT.QUEUE` non esiste, viene creato automaticamente quando il servizio di telemetria (MQXR) viene avviato per la prima volta e impostato per utilizzare l'ID utente guest. Tuttavia, questa attività configura MQ Telemetry per utilizzare uno schema di autorizzazione differente. Per questa attività, creare `SYSTEM.MQTT.TRANSMIT.QUEUE` e configurare l'accesso ad esso, prima di avviare il servizio di telemetria (MQXR).

Esegui il seguente comando:

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

### 3. Impostare la coda di trasmissione predefinita.

È più semplice inviare messaggi direttamente ai client MQTT se `SYSTEM.MQTT.TRANSMIT.QUEUE` è la coda di trasmissione predefinita. Altrimenti, è necessario aggiungere una definizione di coda remota per ogni client che riceve messaggi IBM MQ; consultare [“Invio diretto di un messaggio ad un client” a pagina 312](#). Notare che la modifica della coda di trasmissione predefinita potrebbe interferire con la configurazione esistente.

Quando il servizio di telemetria (MQXR) viene avviato per la prima volta, non imposta `SYSTEM.MQTT.TRANSMIT.QUEUE` come coda di trasmissione predefinita per il gestore code. Per configurare questa impostazione, modificare la proprietà della coda di trasmissione predefinita. A tale scopo, utilizzare IBM MQ Explorer o eseguire il seguente comando:

```
echo ALTER QMGR DEFQXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

### 4. Seguire una procedura in [“Autorizzazione dei client MQTT ad accedere agli oggetti di IBM MQ” a pagina 314](#) per creare uno o più ID utente. Gli ID utente hanno l'autorità di pubblicare, sottoscrivere e inviare pubblicazioni ai client MQTT.

### 5. Installare il servizio di telemetria (MQXR).

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

Vedere anche il codice di esempio in [“Creazione SYSTEM.MQXR.SERVICE” a pagina 310](#).

### 6. Avviare il servizio.

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

Il servizio MQXR (telemetria) viene avviato automaticamente all'avvio del gestore code. Viene avviato manualmente in questa attività, poiché il gestore code è già in esecuzione.

### 7. Utilizzando IBM MQ Explorer, configurare i canali di telemetria per accettare le connessioni dai client MQTT.

I canali di telemetria devono essere configurati in modo che le loro identità siano uno degli ID utente definiti nel passo [“4” a pagina 310](#).

Vedere anche [DEFINE CHANNEL \(MQTT\)](#).

### 8. Verificare la configurazione eseguendo il client di esempio.

Per consentire al client di esempio di utilizzare il canale di telemetria, il canale deve autorizzare il client a pubblicare, sottoscrivere e ricevere pubblicazioni. Il client di esempio si connette al canale di telemetria sulla porta 1883 per impostazione predefinita. Vedere inoltre [IBM MQ Telemetry Transport programmi di esempio](#).

## Creazione SYSTEM.MQXR.SERVICE

Utilizzare il comando `runMQXRService` per creare il `SYSTEM.MQXR.SERVICE`.

V 9.2.4

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+mqxr\bin\runMQXRService.bat') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\" -sf "[DEFAULT]"') +
STOPCMD('+MQ_INSTALL_PATH+mqxr\bin\endMQXRService.bat') +
STOPARG('-m +QMNAME+') +
```

```
STDOUT('+MQ_Q_MGR_DATA_PATH+\mqxr.stdout') +  
STDERR('+MQ_Q_MGR_DATA_PATH+\mqxr.stderr')
```

**Nota:** **V 9.2.4** Da IBM MQ 9.2.4 in poi, l'indicatore **-sf** fornisce il nome file della chiave delle credenziali che contiene la chiave da utilizzare per crittografare le passphrase del canale TLS. Per ulteriori informazioni, vedi [Encryption of passphrase for MQTT TLS channels](#).

Linux

Windows

AIX

## Configurazione dell'accodamento distribuito per inviare messaggi ai client MQTT

Le applicazioni IBM MQ possono inviare i messaggi dei client MQTT v3 pubblicando su una sottoscrizione creata da un client o inviando direttamente un messaggio. Indipendentemente dal metodo utilizzato, il messaggio viene inserito in `SYSTEM.MQTT.TRANSMIT.QUEUE` e inviato al client dal servizio di telemetria (MQXR). Esistono diversi modi per inserire un messaggio su `SYSTEM.MQTT.TRANSMIT.QUEUE`.

### Publicazione di un messaggio in risposta a una sottoscrizione client MQTT

Il servizio di telemetria (MQXR) crea una sottoscrizione per conto del client MQTT. Il client è la destinazione per tutte le pubblicazioni che corrispondono alla sottoscrizione inviata dal client. I servizi di telemetria inoltrano le pubblicazioni corrispondenti al client.

Un client MQTT è connesso a IBM MQ come gestore code, con il proprio nome gestore code impostato su `ClientIdentifier`. La destinazione delle pubblicazioni da inviare al client è una coda di trasmissione, `SYSTEM.MQTT.TRANSMIT.QUEUE`. Il servizio di telemetria inoltra i messaggi su `SYSTEM.MQTT.TRANSMIT.QUEUE` ai client MQTT, utilizzando il nome del Gestore code di destinazione come chiave per un client specifico.

Il servizio di telemetria (MQXR) apre la coda di trasmissione utilizzando `ClientIdentifier` come nome gestore code. Il servizio di telemetria (MQXR) passa l'handle oggetto della coda alla chiamata `MQSUB`, per inoltrare le pubblicazioni che corrispondono alla sottoscrizione client. Nella risoluzione del nome oggetto, il `ClientIdentifier` viene creato come nome del gestore code remoto e la coda di trasmissione deve risolversi in `SYSTEM.MQTT.TRANSMIT.QUEUE`. Utilizzando la risoluzione del nome oggetto IBM MQ standard, `ClientIdentifier` viene risolto come segue; consultare [Tabella 16 a pagina 312](#).

1. `ClientIdentifier` non corrisponde a nulla.

`ClientIdentifier` è un nome gestore code remoto. Non corrisponde al nome del gestore code locale, a un alias del gestore code o a un nome della coda di trasmissione.

Il nome della coda non è definito. Attualmente, il servizio di telemetria (MQXR) imposta `SYSTEM.MQTT.PUBLICATION.QUEUE` come nome della coda. Un client MQTT v3 non supporta le code, quindi il nome della coda risolta viene ignorato dal client.

La proprietà del gestore code locale, Coda di trasmissione predefinita, deve essere impostata su `SYSTEM.MQTT.TRANSMIT.QUEUE`, in modo che la pubblicazione venga inserita in `SYSTEM.MQTT.TRANSMIT.QUEUE` per essere inviata al client.

2. `ClientIdentifier` corrisponde a un alias di gestore code denominato `ClientIdentifier`.

`ClientIdentifier` è un nome gestore code remoto. Corrisponde al nome di un alias del gestore code.

L'alias del gestore code deve essere definito con `ClientIdentifier` come nome del gestore code remoto.

Impostando il nome della coda di trasmissione nella definizione alias del gestore code, non è necessario che la trasmissione predefinita sia impostata su `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Tabella 16. Risoluzione dei nomi di un alias del gestore code MQTT

	Immissione		Output		
<i>ClientIdentifier</i>	Nome del gestore code	Nome coda	Nome del gestore code	Nome coda	Coda di trasmissione
Non corrisponde a nulla	<i>ClientIdentifier</i>	<i>non definito</i>	<i>ClientIdentifier</i>	<i>non definito</i>	Coda di trasmissione predefinita.  SYSTEM.MQTT.TRANSMIT.QUEUE
Corrisponde a un alias del gestore code denominato <i>ClientIdentifier</i>	<i>ClientIdentifier</i>	<i>non definito</i>	<i>ClientIdentifier</i>	<i>non definito</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Per ulteriori informazioni sulla risoluzione dei nomi, consultare [Risoluzione dei nomi](#).

Qualsiasi programma IBM MQ può eseguire la pubblicazione nello stesso argomento. La pubblicazione viene inviata ai relativi sottoscrittori, inclusi i client MQTT v3 che hanno una sottoscrizione all'argomento.

Se un argomento di gestione viene creato in un cluster, con l'attributo `CLUSTER(clusterName)`, qualsiasi applicazione nel cluster può essere pubblicata sul client; ad esempio:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

Figura 16. Definizione di un argomento cluster su Windows

**Nota:** Non fornire a SYSTEM.MQTT.TRANSMIT.QUEUE un attributo cluster.

I publisher e i sottoscrittori del client MQTT possono connettersi a diversi gestori code. I sottoscrittori e i publisher possono far parte dello stesso cluster o essere connessi da una gerarchia di pubblicazione / sottoscrizione. La pubblicazione viene consegnata dal publisher al sottoscrittore utilizzando IBM MQ.

## Invio diretto di un messaggio ad un client

Un'alternativa a un client che crea una sottoscrizione e riceve una pubblicazione che corrisponde all'argomento della sottoscrizione, invia un messaggio direttamente a un client MQTT v3. Le applicazioni client MQTT v3 non possono inviare messaggi direttamente, ma altre applicazioni, come le applicazioni IBM MQ, possono farlo.

L'applicazione IBM MQ deve conoscere il `ClientIdentifier` del client MQTT v3. Poiché i client MQTT v3 non hanno code, il nome della coda di destinazione viene passato al metodo MQTT v3 application `client messageArrived` come un nome argomento. Ad esempio, in un programma MQI, creare un descrittore oggetto con il client come `ObjectQmgrName`:

```
MQOD.ObjectQmgrName = ClientIdentifier ;
MQOD.ObjectName = name ;
```

Figura 17. Descrittore oggetto MQI per inviare un messaggio a una destinazione client MQTT v3

Se l'applicazione viene scritta utilizzando JMS, creare una destinazione point-to-point; ad esempio:

```
javax.jms.Destination jmsDestination =  
(javax.jms.Destination)jmsFactory.createQueue  
("queue://ClientIdentifier/name");
```

Figura 18. JMS destinazione per l'invio di un messaggio a un client MQTT v3

Per inviare un messaggio non richiesto a un client MQTT , utilizzare una definizione di coda remota. Il nome del gestore coda remoto deve essere risolto nel ClientIdentifier del client. La coda di trasmissione deve essere risolta in SYSTEM.MQTT.TRANSMIT.QUEUE ; consultare Tabella 17 a pagina 313. Il nome della coda remota può essere qualsiasi cosa. Il client lo riceve come stringa di argomenti.

Tabella 17. Risoluzione dei nomi di una definizione di coda remota del client MQTT

Immissione		Output		
Nome coda	Nome del gestore code	Nome coda	Nome del gestore code	Coda di trasmissione
Nome della definizione della coda remota	Nome gestore code locale o vuoto	Nome coda remota utilizzato come stringa argomento	ClientIdentifier	SYSTEM.MQTT.TRANSMIT.QUEUE

Se il client è connesso, il messaggio viene inviato direttamente al client MQTT , che richiama il metodo messageArrived ; consultare [messageArrived method](#).

Se il client si è scollegato da una sessione persistente, il messaggio viene memorizzato in SYSTEM.MQTT.TRANSMIT.QUEUE ; consultare [MQTT Sessioni stateless e stateful](#). Viene inoltrato al client quando il client si riconnette di nuovo alla sessione.

Se si invia un messaggio non persistente, questo viene inviato al client con al massimo QoS (quality of service) QoS=0. Se si invia un messaggio persistente direttamente a un client, per impostazione predefinita, viene inviato con esattamente una volta QoS (quality of service), QoS=2. Poiché il client potrebbe non avere un meccanismo di persistenza, il client può ridurre la qualità del servizio che accetta per i messaggi inviati direttamente. Per ridurre la qualità del servizio per i messaggi inviati direttamente a un client, effettuare una sottoscrizione all'argomento DEFAULT . QoS. Specificare la qualità massima del servizio che il client può supportare.

Linux

Windows

AIX

## Autenticazione, autorizzazione e identificazione del client MQTT

Il servizio di telemetria (MQXR) pubblica o sottoscrive gli argomenti IBM MQ per conto dei client MQTT , utilizzando i canali MQTT . L'amministratore IBM MQ configura l'identità del canale MQTT utilizzata per l'autorizzazione IBM MQ . L'amministratore può definire un'identità comune per il canale oppure utilizzare il Nome utente o ClientIdentifier di un client connesso al canale.

Il servizio di telemetria (MQXR) può autenticare il client utilizzando il Nome utente fornito dal client o utilizzando un certificato client. Il Nome utente viene autenticato utilizzando una password fornita dal client.

Per riassumere: l'identificazione del client è la selezione dell'identità del client. In base al contesto, il client viene identificato da ClientIdentifier, Username, un'identità client comune creata dall'amministratore o un certificato client. L'identificativo client utilizzato per il controllo di autenticità non deve essere lo stesso identificativo utilizzato per l'autorizzazione.

I programmi client MQTT impostano Nome utente e Password che vengono inviati al server utilizzando un canale MQTT . Possono anche impostare le proprietà TLS richieste per codificare e autenticare la connessione. L'amministratore decide se autenticare il canale MQTT e come autenticare il canale.

Per autorizzare un client MQTT ad accedere agli oggetti IBM MQ, autorizzare ClientIdentificario Username del client oppure autorizzare un'identità client comune. Per consentire a un client di connettersi a IBM MQ, autenticare il Nome utenteo utilizzare un certificato client. Configurare JAAS per autenticare il Nome utentee configurare TLS per autenticare un certificato client.

Se imposti una Password sul client, crittografa la connessione utilizzando VPN o configura il canale MQTT per utilizzare TLS, per mantenere la password privata.

È difficile gestire i certificati client. Per questo motivo, se i rischi associati all'autenticazione della password sono accettabili, l'autenticazione della password viene spesso utilizzata per autenticare i client.

Se esiste un modo sicuro per gestire e memorizzare il certificato client, è possibile fare affidamento sull'autenticazione del certificato. Tuttavia, raramente i certificati possono essere gestiti in modo sicuro nei tipi di ambienti in cui viene utilizzata la telemetria. Invece, l'autenticazione delle unità che utilizzano i certificati client è completata dall'autenticazione delle password client sul server. A causa della complessità aggiuntiva, l'utilizzo dei certificati client è limitato alle applicazioni altamente sensibili. L'uso di due forme di autenticazione è chiamato autenticazione a due fattori. È necessario conoscere uno dei fattori, ad esempio una password, e l'altro, ad esempio un certificato.

In un'applicazione altamente sensibile, ad esempio un dispositivo con chip e pin, il dispositivo viene bloccato durante la fabbricazione per evitare la manomissione dell'hardware e del software interni. Un certificato client sicuro, limitato nel tempo, viene copiato sul dispositivo. Il dispositivo viene distribuito nell'ubicazione in cui deve essere utilizzato. Un'ulteriore autenticazione viene eseguita ogni volta che il dispositivo viene utilizzato, utilizzando una parola d'ordine o un altro certificato da una smart card.

## Linux Windows AIX **Identità e autorizzazione del client MQTT**

Utilizzare l'ID client, Nome utente, o un'identità client comune per l'autorizzazione ad accedere agli oggetti IBM MQ.

L'amministratore IBM MQ ha tre opzioni per selezionare l'identità del canale MQTT. L'amministratore effettua la scelta quando definisce o modifica il canale MQTT utilizzato dal client. L'identità viene utilizzata per autorizzare l'accesso agli argomenti IBM MQ. La scelta viene effettuata nel seguente ordine:

1. L'ID client (vedere [USECLNTID](#)).
2. Un'identità fornita dall'amministratore per il canale (MCAUSER del canale. Vedere [MCAUSER](#)).
3. Se nessuna delle scelte precedenti si applica, il Nome utente passato dal client MQTT (Nome utente è un attributo della classe `MqttConnectOptions`. Deve essere impostato prima che il client si colleghi al servizio. Il valore predefinito è null).

**Prevenzione dei problemi:** L'identità scelta da questo processo viene successivamente indicata, ad esempio dal comando `DISPLAY CHSTATUS (MQTT)`, come MCAUSER del client. Tenere presente che questa non è necessariamente la stessa identità del MCAUSER del canale a cui si fa riferimento nella scelta (2).

Utilizzare il comando IBM MQ **setmqaut** per selezionare quali oggetti e quali azioni sono autorizzati ad essere utilizzati dall'identit. associata al canale MQTT. Ad esempio, il seguente codice autorizza un'identità del canale `MQTTClient`, fornita dall'amministratore del gestore code `QM1`:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

## Linux Windows AIX **Autorizzazione dei client MQTT ad accedere agli oggetti di IBM MQ**

Attenersi alla seguente procedura per autorizzare i client di MQTT a pubblicare e sottoscrivere oggetti IBM MQ. La procedura segue quattro modelli di controllo accessi alternativi.

## Prima di iniziare

I client MQTT sono autorizzati ad accedere agli oggetti in IBM MQ mediante l'assegnazione di un'identità quando si connettono a un canale di telemetria. L'amministratore IBM MQ configura il canale di telemetria utilizzando IBM MQ Explorer per fornire a un client uno dei seguenti tre tipi di identità:

1. `ClientIdentifier`
2. Nome utente
3. Un nome che l'amministratore assegna al canale.

Indipendentemente dal tipo utilizzato, l'identità deve essere definita in IBM MQ come principal dal servizio di autorizzazione installato. Il servizio di autorizzazione predefinito su Windows o Linux è denominato OAM (Object Authority Manager). Se si sta utilizzando OAM, l'identità deve essere definita come un ID utente.

Utilizzare l'identità per fornire a un client, o a una raccolta di client, l'autorizzazione a pubblicare o sottoscrivere argomenti definiti in IBM MQ. Se un client MQTT ha sottoscritto un argomento, utilizzare l'identità per fornire l'autorizzazione a ricevere le pubblicazioni risultanti.

È difficile gestire un sistema con decine di migliaia di client MQTT, ognuno dei quali richiede autorizzazioni di accesso individuali. Una soluzione è definire identità comuni e associare singoli client MQTT a una delle identità comuni. Definire tutte le identità comuni necessarie per definire diverse combinazioni di autorizzazioni. Un'altra soluzione è quella di scrivere il proprio servizio di autorizzazione che può trattare più facilmente con migliaia di utenti rispetto al sistema operativo.

È possibile combinare i client MQTT in identità comuni in due modi, utilizzando OAM:

1. Definire più canali di telemetria, ciascuno con un diverso ID utente assegnato dall'amministratore tramite Esplora risorse di IBM MQ. I client che si collegano utilizzando numeri di porta TCP/IP differenti sono associati a canali di telemetria diversi e sono assegnati a identità differenti.
2. Definire un singolo canale di telemetria, ma far selezionare a ciascun client un Nome utente da una piccola serie di ID utente. L'amministratore configura il canale di telemetria per selezionare il Nome utente del client come sua identità.

In questa attività, l'identità del canale di telemetria è denominata *mqttUser*, indipendentemente da come è impostata. Se le raccolte di client utilizzano identità differenti, utilizzare più *mqttUsers*, uno per ogni raccolta di client. Poiché l'attività utilizza OAM, ogni *mqttUser* deve essere un ID utente.

## Informazioni su questa attività

In questa attività, è possibile scegliere tra quattro pattern di controllo accessi che è possibile personalizzare in base a requisiti specifici. I modelli differiscono nella loro granularità del controllo accessi.

- [“Nessun controllo accessi” a pagina 315](#)
- [“Controllo degli accessi generici” a pagina 316](#)
- [“Controllo accessi a grana media” a pagina 316](#)
- [“Controllo dell'accesso dettagliato” a pagina 316](#)

Il risultato dei modelli consiste nell'assegnare a *mqttUsers* serie di autorizzazioni per pubblicare e sottoscrivere IBM MQ e ricevere pubblicazioni da IBM MQ.

### *Nessun controllo accessi*

Ai client MQTT viene fornita l'autorizzazione di gestione IBM MQ e può eseguire qualsiasi azione su qualsiasi oggetto.

## Procedura

1. Creare un ID utente *mqttUser* per agire come identità di tutti i client MQTT.

2. Aggiungere *mqttUser* al gruppo *mqm* ; consultare [Aggiunta di un utente a un gruppo su Windows o Creazione e gestione di gruppi su Linux](#)

#### Controllo degli accessi generici

I client MQTT hanno l'autorità per pubblicare e sottoscrivere e per inviare messaggi a client MQTT . Non dispongono dell'autorità per eseguire altre azioni o per accedere ad altri oggetti.

### Procedura

1. Creare un ID utente *mqttUser* per agire come identità di tutti i client MQTT .
2. Autorizzare *mqttUser* a pubblicare e sottoscrivere tutti gli argomenti e a inviare pubblicazioni ai client MQTT .

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub  
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

#### Controllo accessi a grana media

I client MQTT sono divisi in gruppi differenti per pubblicare e sottoscrivere diverse serie di argomenti e per inviare messaggi a client MQTT .

### Procedura

1. Creare più ID utente, *mqttUser* e più argomenti di gestione nella struttura ad albero degli argomenti di pubblicazione / sottoscrizione.
2. Autorizzare *mqttUsers* differenti per argomenti differenti.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub  
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. Creare un gruppo *mqtt* aggiungere tutti i *mqttUsers* al gruppo.
4. Autorizzare *mqtt* a inviare argomenti ai client MQTT .

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

#### Controllo dell'accesso dettagliato

I client MQTT sono integrati in un sistema di controllo accessi esistente, che autorizza i gruppi a eseguire azioni sugli oggetti.

### Informazioni su questa attività

Un ID utente viene assegnato a uno o più gruppi di sistemi operativi a seconda delle autorizzazioni richieste. Se le applicazioni IBM MQ stanno eseguendo la pubblicazione e la sottoscrizione allo stesso spazio argomento dei client MQTT , utilizzare questo modello. I gruppi vengono indicati come Publish X, Subscribe Y e mqtt

#### Publish X

I membri dei gruppi Publish X possono pubblicare in *topicX*.

#### Subscribe Y

I membri dei gruppi Subscribe Y possono sottoscrivere *topicY*.

#### mqtt

I membri del gruppo *mqtt* possono inviare pubblicazioni ai client MQTT .

### Procedura

1. Creare più gruppi, Publish X e Subscribe Y , assegnati a più argomenti di gestione nella struttura ad albero degli argomenti di pubblicazione / sottoscrizione.

2. Creare un gruppo `mqtt`.
3. Creare più ID utente, `mqttUsers`, e aggiungere gli utenti a uno qualsiasi dei gruppi, a seconda di ciò che sono autorizzati a fare.
4. Autorizzare diversi gruppi `Publish X` e `Subscribe X` a diversi argomenti e autorizzare il gruppo `mqtt` a inviare messaggi ai client MQTT.

```
setmqaut -m qMgr -t topic -n topic1 -p Publish X -all +pub
setmqaut -m qMgr -t topic -n topic1 -p Subscribe X -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

## Linux Windows AIX Autenticazione client MQTT utilizzando una parola d'ordine

Autenticare il Nome utente utilizzando la password client. È possibile autenticare il client utilizzando un'identità diversa rispetto all'identità utilizzata per autorizzare il client a pubblicare e sottoscrivere argomenti.

Il servizio di telemetria (MQXR) utilizza JAAS per autenticare il client Username. JAAS utilizza Password fornita dal client MQTT.

L'amministratore IBM MQ decide se autenticare il Nome utente o non autenticarlo affatto, configurando il canale MQTT a cui si connette un client. I client possono essere assegnati a canali diversi e ciascun canale può essere configurato per autenticare i propri client in modi differenti. Utilizzando JAAS, è possibile configurare quali metodi devono autenticare il client e quali possono facoltativamente autenticare il client.

La scelta dell'identità per l'autenticazione non influisce sulla selezione dell'identità per l'autorizzazione. È possibile impostare un'identità comune per l'autorizzazione per comodità di gestione, ma autenticare ciascun utente per utilizzare tale identità. La seguente procedura descrive la procedura per autenticare i singoli utenti per utilizzare un'identità comune:

1. L'amministratore IBM MQ imposta l'identità del canale MQTT su qualsiasi nome, ad esempio `MQTTClientUser`, utilizzando Esplora risorse di IBM MQ.
2. L'amministratore IBM MQ autorizza `MQTTClient` a pubblicare e sottoscrivere qualsiasi argomento:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. Lo sviluppatore dell'applicazione client MQTT crea un oggetto `MqttConnectOptions` e imposta Nome utente e Password prima di connettersi al server.
4. Lo sviluppatore della sicurezza crea un JAAS `LoginModule` per autenticare il nome utente con Password e lo include nel file di configurazione JAAS.
5. L'amministratore IBM MQ configura il canale MQTT per autenticare il Username del client utilizzando JAAS.

## Linux Windows AIX Autenticazione client MQTT mediante TLS

Le connessioni tra il client MQTT e il gestore code vengono sempre avviate dal client MQTT. Il client MQTT è sempre il client SSL. L'autenticazione client del server e l'autenticazione server del client MQTT sono entrambe facoltative.

Fornire al client un certificato digitale firmato privato, è possibile autenticare il client MQTT su WebSphere MQ. L'amministratore di WebSphere MQ può forzare i client MQTT ad autenticarsi nel gestore code utilizzando TLS. È possibile richiedere l'autenticazione client solo come parte dell'autenticazione reciproca.

Come alternativa all'utilizzo di SSL, alcuni tipi di VPN (Virtual Private Network), ad esempio IPsec, autenticano gli endpoint di una connessione TCP/IP. VPN codifica ogni pacchetto IP che passa attraverso

la rete. Una volta che tale connessione VPN è stabilita, è stata creata una rete attendibile. È possibile connettere i client MQTT ai canali di telemetria utilizzando TCP/IP sulla rete VPN.

L'autenticazione client tramite TLS si basa sulla presenza di un segreto nel client. Il segreto è la chiave privata del client nel caso di un certificato autofirmato o una chiave fornita da un'autorità di certificazione. La chiave viene utilizzata per firmare il certificato digitale del client. Chiunque sia in possesso della corrispondente chiave pubblica può verificare il certificato digitale. I certificati possono essere attendibili o, se sono concatenati, è possibile risalire attraverso una catena di certificati a un certificato root attendibile. La verifica del client invia al server tutti i certificati presenti nella catena di certificati fornita dal client. Il server controlla la catena di certificati finché non trova un certificato attendibile. Il certificato attendibile è il certificato pubblico generato da un certificato autofirmato oppure un certificato root in genere emesso da un'autorità di certificazione. Come ultimo passo facoltativo, il certificato attendibile può essere confrontato con un elenco di revocche di certificati in tempo reale.

Il certificato attendibile potrebbe essere emesso da un'autorità di certificazione ed essere già incluso nell'archivio certificati JRE. Potrebbe essere un certificato autofirmato oppure un certificato che è stato aggiunto al keystore del canale di telemetria come certificato attendibile.

**Nota:** Il canale di telemetria include un keystore/truststore combinato che contiene sia le chiavi private per uno o più canali di telemetria, sia eventuali certificati pubblici necessari per autenticare i client. Poiché un canale SSL deve disporre di un keystore ed è lo stesso file del truststore del canale, non viene mai fatto riferimento all'archivio certificati JRE. L'implicazione è che se l'autenticazione di un client richiede un certificato root CA, è necessario posizionare il certificato root nel keystore per il canale, anche se il certificato root CA si trova già nell'archivio certificati JRE. All'archivio certificati JRE non viene mai fatto riferimento.

Occorre considerare le minacce che l'autenticazione client deve contrastare e i ruoli che il client e il server ricoprono nel contrastare le minacce. L'autenticazione del certificato client da sola non è sufficiente a impedire l'accesso non autorizzato a un sistema. Se qualcun altro ha accesso al dispositivo client, tale dispositivo non effettua necessariamente azioni autorizzate dal titolare del certificato. Non fare mai affidamento su una singola difesa contro attacchi indesiderati. Utilizzare almeno un approccio di autenticazione a due fattori e integrare il possesso di un certificato con la conoscenza di informazioni private. Ad esempio, utilizzare JAAS e autenticare il client utilizzando una password emessa dal server.

La minaccia principale al certificato client è che esso cada nelle mani sbagliate. Il certificato è contenuto in un keystore protetto da password nel client. Come viene collocato nel keystore? In che modo il client MQTT ottiene la password sul keystore? Quanto è sicura la protezione con password? I dispositivi di telemetria sono spesso facili da rimuovere e possono essere oggetto di attacchi in privato. Il dispositivo deve essere a prova di manomissione? La distribuzione e la protezione dei certificati sul lato client è notoriamente difficile: si chiama problema di gestione chiavi.

Una minaccia secondaria è che il dispositivo venga utilizzato erroneamente per accedere ai server in modi non intenzionali. Ad esempio, se l'applicazione MQTT viene manomessa, potrebbe essere possibile utilizzare un punto debole della configurazione del server utilizzando l'identità del client autenticato.

Per autenticare un client MQTT tramite SSL, configurare il canale di telemetria e il client.

### **Concetti correlati**

[“Configurazione del canale di telemetria per l'autenticazione client MQTT utilizzando TLS” a pagina 318](#)

L'amministratore IBM MQ configura i canali di telemetria sul server. Ogni canale è configurato per accettare una connessione TCP/IP su un numero di porta diverso. I canali TLS sono configurati con l'accesso protetto da passphrase ai file chiave. Se un canale TLS è definito senza passphrase o file chiave, il canale non accetta le connessioni TLS.

### **Attività correlate**

[Configurazione client MQTT per l'autenticazione client utilizzando TLS](#)

## **Configurazione del canale di telemetria per l'autenticazione client MQTT utilizzando TLS**

L'amministratore IBM MQ configura i canali di telemetria sul server. Ogni canale è configurato per accettare una connessione TCP/IP su un numero di porta diverso. I canali TLS sono configurati con



l'accesso protetto da passphrase ai file chiave. Se un canale TLS è definito senza passphrase o file chiave, il canale non accetta le connessioni TLS.

Impostare la proprietà `com.ibm.mq.MQTT.ClientAuth` di un canale di telemetria TLS su `REQUIRED` per forzare tutti i client che si connettono su tale canale a fornire la prova che hanno verificato i certificati digitali. I certificati client vengono autenticati utilizzando i certificati dalle autorità di certificazione, che portano a un certificato root attendibile. Se il certificato client è autofirmato o è firmato da un certificato che proviene da un'autorità di certificazione, i certificati firmati pubblicamente del client o dell'autorità di certificazione devono essere memorizzati in modo sicuro sul server.

Inserire il certificato client firmato pubblicamente o il certificato dall'autorità di certificazione nel keystore del canale di telemetria. Sul server, i certificati firmati pubblicamente vengono memorizzati nello stesso file chiave dei certificati firmati privatamente, piuttosto che in un truststore separato.

Il server verifica la firma di tutti i certificati client che vengono inviati utilizzando tutti i certificati pubblici e le suite di cifratura di cui dispone. Il server verifica la catena di chiavi. Il gestore code può essere configurato per verificare il certificato rispetto all'elenco di revoca del certificato. La proprietà dell'elenco nomi di revoca del gestore code è `SSLCRLNL`.

Se uno qualsiasi dei certificati inviati da un client viene verificato da un certificato nel keystore del server, il client viene autenticato.

L'amministratore IBM MQ può configurare lo stesso canale di telemetria per utilizzare JAAS per controllare `UserName` o `ClientIdentifier` del client con la `Password` del client.

È possibile utilizzare lo stesso keystore per più canali di telemetria.

La verifica di almeno un certificato digitale nel keystore del client protetto da password sul dispositivo autentica il client sul server. Il certificato digitale viene utilizzato solo per l'autenticazione da IBM MQ. Non viene utilizzato per verificare l'indirizzo TCP/IP del client o per impostare l'identità del client per l'autorizzazione o l'account. L'identità del client adottato dal server è il `Nome utente` o `ClientIdentifier` del client oppure un'identità creata dall'amministratore IBM MQ.

Puoi anche utilizzare le suite di cifratura TLS per l'autenticazione client. Se si prevede di utilizzare le suite di cifratura SHA-2, consultare [“Requisiti di sistema per l'utilizzo delle suite di crittografia SHA-2 con i canali MQTT”](#) a pagina 320.

### Concetti correlati

[“Configurazione del canale di telemetria per l'autenticazione di canale utilizzando TLS”](#) a pagina 320

L'amministratore IBM MQ configura i canali di telemetria sul server. Ogni canale è configurato per accettare una connessione TCP/IP su un numero di porta diverso. I canali TLS sono configurati con l'accesso protetto da passphrase ai file chiave. Se un canale TLS è definito senza passphrase o file chiave, il canale non accetta le connessioni TLS.

### Riferimenti correlati

[DEFINE CHANNEL \(MQTT\)](#)

[MODIFICA CANALE \(MQTT\)](#)

[CipherSpecs e CipherSuites](#)

Linux

Windows

AIX

## Autenticazione del canale di telemetria mediante TLS

Le connessioni tra il client MQTT e il gestore code vengono sempre avviate dal client MQTT. Il client MQTT è sempre il client SSL. L'autenticazione client del server e l'autenticazione server del client MQTT sono entrambe facoltative.

Il client tenta sempre di autenticare il server, a meno che il client non sia configurato in modo da utilizzare una `CipherSpec` che supporta la connessione anonima. Se l'autenticazione non riesce, la connessione non viene stabilita.

Come alternativa all'utilizzo di SSL, alcuni tipi di VPN (Virtual Private Network), ad esempio IPsec, autenticano gli endpoint di una connessione TCP/IP. VPN codifica ogni pacchetto IP che passa attraverso

la rete. Una volta che tale connessione VPN è stabilita, è stata creata una rete attendibile. È possibile connettere i client MQTT ai canali di telemetria utilizzando TCP/IP sulla rete VPN.

L'autenticazione server tramite SSL autentica il server al quale si sta per inviare informazioni riservate. Il client esegue le verifiche che corrispondono ai certificati inviati dal server, rispetto ai certificati collocati nel relativo truststore o nel relativo cacerts archivio JRE.

L'archivio certificati JRE è un file JKS, cacerts. Si trova in JRE InstallPath\lib\security\ . È installato con la password predefinita changeit. È possibile memorizzare certificati attendibili nell'archivio certificati JRE o nel truststore del client. Non è possibile utilizzare entrambi gli archivi. Utilizzare il truststore client se si desidera tenere i certificati pubblici che il client ritiene attendibili separati dai certificati utilizzati da altre applicazioni Java. Utilizzare l'archivio certificati JRE se si desidera utilizzare un archivio certificati comune per tutte le applicazioni Java in esecuzione sul client. Se si decide di utilizzare l'archivio certificati JRE, riesaminare i certificati in esso contenuti per assicurarsi che siano attendibili.

È possibile modificare la configurazione JSSE fornendo un diverso provider di trust. È possibile personalizzare un provider di trust per eseguire diversi controlli su un certificato. In alcuni ambienti OGSi che hanno utilizzato il client MQTT, l'ambiente fornisce un provider di attendibilità differente.

Per autenticare il canale di telemetria tramite TLS, configurare il server e il client.

## Linux Windows AIX **Configurazione del canale di telemetria per l'autenticazione di canale utilizzando TLS**

L'amministratore IBM MQ configura i canali di telemetria sul server. Ogni canale è configurato per accettare una connessione TCP/IP su un numero di porta diverso. I canali TLS sono configurati con l'accesso protetto da passphrase ai file chiave. Se un canale TLS è definito senza passphrase o file chiave, il canale non accetta le connessioni TLS.

Memorizzare il certificato digitale del server, firmato con la sua chiave privata, nel keystore che il canale di telemetria utilizzerà sul server. Memorizzare i certificati nella relativa catena di chiavi nel keystore, se si desidera trasmettere la catena di chiavi al client. Configurare il canale di telemetria utilizzando IBM MQ explorer per utilizzare TLS. Fornire il percorso al keystore e la passphrase per accedere al keystore. Se non si imposta il numero di porta TCP/IP del canale, il numero di porta del canale di telemetria TLS assume il valore predefinito 8883.

Puoi anche utilizzare le suite di cifratura TLS per l'autenticazione del canale. Se si prevede di utilizzare le suite di cifratura SHA-2 , consultare [“Requisiti di sistema per l'utilizzo delle suite di crittografia SHA-2 con i canali MQTT”](#) a pagina 320.

### **Concetti correlati**

[“Configurazione del canale di telemetria per l'autenticazione client MQTT utilizzando TLS”](#) a pagina 318

L'amministratore IBM MQ configura i canali di telemetria sul server. Ogni canale è configurato per accettare una connessione TCP/IP su un numero di porta diverso. I canali TLS sono configurati con l'accesso protetto da passphrase ai file chiave. Se un canale TLS è definito senza passphrase o file chiave, il canale non accetta le connessioni TLS.

### **Riferimenti correlati**

[DEFINISCI CANALE \(MQTT\)](#)

[MODIFICA CANALE \(MQTT\)](#)

[CipherSpecs e CipherSuites](#)

## Linux Windows AIX **Requisiti di sistema per l'utilizzo delle suite di crittografia SHA-2 con i canali MQTT**

Se si utilizza una versione di Java che supporta le suite di crittografia SHA-2 , è possibile utilizzare queste suite per proteggere i canali MQTT (telemetria) e le applicazioni client.

Per IBM MQ 8.0 , che comprende il servizio di telemetria (MQXR), la versione minima di Java è Java 7 da IBM , SR6. Le suite di cifratura SHA-2 sono supportate per impostazione predefinita in Java 7 da IBM, SR4

in poi. È quindi possibile utilizzare le suite di crittografia SHA-2 con il servizio di telemetria (MQXR) per proteggere i canali MQTT (telemetria).

Se si sta eseguendo un client MQTT con un JRE differente, è necessario assicurarsi che supporti anche le suite di cifratura SHA-2 .

### Concetti correlati

[Servizio di telemetria \(MQXR\)](#)

[“Configurazione del canale di telemetria per l'autenticazione di canale utilizzando TLS” a pagina 320](#)

L'amministratore IBM MQ configura i canali di telemetria sul server. Ogni canale è configurato per accettare una connessione TCP/IP su un numero di porta diverso. I canali TLS sono configurati con l'accesso protetto da passphrase ai file chiave. Se un canale TLS è definito senza passphrase o file chiave, il canale non accetta le connessioni TLS.

### Riferimenti correlati

[DEFINE CHANNEL \(MQTT\)](#)

[MODIFICA CANALE \(MQTT\)](#)

Linux

Windows

AIX

## Privacy di pubblicazione sui canali di telemetria

La riservatezza delle pubblicazioni MQTT inviate in entrambe le direzioni attraverso i canali di telemetria è protetta utilizzando TLS per codificare le trasmissioni sulla connessione.

I client MQTT che si collegano ai canali di telemetria utilizzano TLS per proteggere la privacy delle pubblicazioni trasmesse sul canale utilizzando la codifica a chiave simmetrica. Poiché gli endpoint non sono autenticati, non puoi considerare attendibile la sola crittografia del canale. Combina la protezione della privacy con il server o l'autenticazione reciproca.

Come alternativa all'utilizzo di SSL, alcuni tipi di VPN (Virtual Private Network), ad esempio IPsec, autenticano gli endpoint di una connessione TCP/IP. VPN codifica ogni pacchetto IP che passa attraverso la rete. Una volta che tale connessione VPN è stabilita, è stata creata una rete attendibile. È possibile connettere i client MQTT ai canali di telemetria utilizzando TCP/IP sulla rete VPN.

Per una configurazione tipica, che codifica il canale e autentica il server, consultare [“Autenticazione del canale di telemetria mediante TLS” a pagina 319](#).

La crittografazione delle connessioni TLS senza autenticare il server espone la connessione agli attacchi man - in - the - middle. Anche se le informazioni scambiate sono protette contro le intercettazioni, non sai con chi le stai scambiando. A meno che tu non controlli la rete, sei esposto a qualcuno che intercetta le tue trasmissioni IP e si maschera come l'endpoint.

Puoi creare una connessione TLS crittografata, senza autenticare il server, utilizzando uno scambio di chiavi Diffie - Hellman CipherSpec che supporti TLS anonimo. Il segreto principale, condiviso tra client e server e utilizzato per codificare le trasmissioni TLS, viene stabilito senza scambiare un certificato server firmato privatamente.

Poiché le connessioni anonime non sono sicure, la maggior parte delle implementazioni TLS non utilizzano per impostazione predefinita CipherSpecanonime. Se una richiesta client per la connessione TLS viene accettata da un canale di telemetria, il canale deve avere un keystore protetto da una passphrase. Per impostazione predefinita, poiché le implementazioni TLS non utilizzano CipherSpecanonime, il keystore deve contenere un certificato firmato privatamente che il client può autenticare.

Se si utilizza CipherSpecanonimo, il keystore del server deve esistere, ma non deve contenere alcun certificato firmato privatamente.

Un altro modo per stabilire una connessione crittografata consiste nel sostituire il provider di attendibilità sul client con la propria implementazione. Il provider di attendibilità non autentica il certificato del server, ma la connessione viene codificata.



**Attenzione:** Quando si utilizza TLS con MQTT è possibile utilizzare messaggi di grandi dimensioni, tuttavia, potrebbe verificarsi un possibile impatto sulle prestazioni. MQTT è ottimizzato per l'elaborazione di piccoli messaggi (in genere tra 1KB e 1MB di dimensione).

## client MQTT Java

Configurare TLS per autenticare il canale di telemetria e il client MQTT Java e codificare il trasferimento dei messaggi tra di essi. MQTT I client Java utilizzano JSSE (Java Secure Socket Extension) per collegare i canali di telemetria utilizzando TLS. Come alternativa all'utilizzo di SSL, alcuni tipi di VPN (Virtual Private Network), ad esempio IPsec, autenticano gli endpoint di una connessione TCP/IP. VPN codifica ogni pacchetto IP che passa attraverso la rete. Una volta che tale connessione VPN è stabilita, è stata creata una rete attendibile. È possibile connettere i client MQTT ai canali di telemetria utilizzando TCP/IP sulla rete VPN.

È possibile configurare la connessione tra un client Java MQTT e un canale di telemetria per utilizzare il protocollo TLS su TCP/IP. Ciò che è protetto dipende dal modo in cui si configura TLS per utilizzare JSSE. A partire dalla configurazione più sicura, è possibile configurare tre diversi livelli di sicurezza:

1. Consentire la connessione solo ai client MQTT attendibili. Connetti un client MQTT solo a un canale di telemetria attendibile. Crittografare i messaggi tra il client e il gestore code; consultare [“Autenticazione client MQTT mediante TLS”](#) a pagina 317
2. Connetti un client MQTT solo a un canale di telemetria attendibile. Crittografare i messaggi tra il client e il gestore code; consultare [“Autenticazione del canale di telemetria mediante TLS”](#) a pagina 319.
3. Crittografare i messaggi tra il client e il gestore code; consultare [“Privacy di pubblicazione sui canali di telemetria”](#) a pagina 321.

## Parametri di configurazione JSSE

Modificare i parametri JSSE per modificare il modo in cui è configurata una connessione TLS. I parametri di configurazione JSSE sono disposti in tre serie:

1. [MQ Telemetry canale](#)
2. [Client MQTT Java](#)
3. [JRE](#)

Configurare i parametri del canale di telemetria utilizzando IBM MQ Explorer. Impostare i parametri MQTT Java Client nell'attributo `MqttConnectionOptions.SSLProperties`. Modificare i parametri di sicurezza JRE modificando i file nella directory di sicurezza JRE sul client e sul server.

### MQ Telemetry canale

Impostare tutti i parametri TLS del canale di telemetria utilizzando IBM MQ Explorer.

#### ChannelName

`ChannelName` è un parametro obbligatorio su tutti i canali.

Il nome del canale identifica il canale associato ad un numero di porta particolare. Denominare i canali per gestire le serie di client MQTT.

#### PortNumber

`PortNumber` è un parametro facoltativo su tutti i canali. Il valore predefinito è 1883 per i canali TCP e 8883 per i canali TLS.

Il numero di porta TCP/IP associato a questo canale. I client MQTT sono connessi ad un canale specificando la porta definita per il canale. Se il canale ha proprietà TLS, il client deve connettersi utilizzando il protocollo TLS; ad esempio:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId1");
mqttClient.connect();
```

#### NomeKeyFile

`KeyFileName` è un parametro obbligatorio per i canali TLS. Deve essere omesso per i canali TCP.

KeyFileNome è il percorso del keystore Java contenente certificati digitali forniti. Utilizzare JKS, JCEKS o PKCS12 come tipo di keystore sul server.

Identificare il tipo di keystore utilizzando una delle seguenti estensioni file:

- .jks
- .jceks
- .p12
- .pkcs12

Si presume che un keystore con qualsiasi altra estensione file sia un keystore JKS.

È possibile combinare un tipo di keystore sul server con altri tipi di keystore sul client.

Inserire il certificato privato del server nel keystore. Il certificato è noto come certificato server. Il certificato può essere autofirmato o parte di una concatenazione di certificati firmata da un'autorità di firma.

Se si sta utilizzando una catena di certificati, inserire i certificati associati nel keystore del server.

Il certificato del server e tutti i certificati nella sua catena di certificati vengono inviati ai client per autenticare l'identità del server.

Se ClientAuth è stato impostato su Required, il keystore deve contenere i certificati necessari per autenticare il client. Il client invia un certificato autofirmato, o una catena di certificati, e il client viene autenticato dalla prima verifica di questo materiale rispetto a un certificato nel keystore. Utilizzando una catena di certificati, un certificato può verificare molti client, anche se vengono emessi con certificati client differenti.

### **PassPhrase**

PassPhrase è un parametro obbligatorio per i canali TLS. Deve essere omesso per i canali TCP.

La passphrase viene utilizzata per proteggere il keystore.

### **ClientAuth**

ClientAuth è un parametro TLS facoltativo. L'impostazione predefinita è nessuna autenticazione client. Deve essere omesso per i canali TCP.

Impostare ClientAuth se si desidera che il servizio di telemetria (MQXR) autentichi il client, prima di consentire al client di connettersi al canale di telemetria.

Se si imposta ClientAuth, il client deve connettersi al server utilizzando TLS e autenticare il server. In risposta all'impostazione di ClientAuth, il client invia il proprio certificato digitale al server e qualsiasi altro certificato nel relativo keystore. Il suo certificato digitale è noto come certificato client. Questi certificati vengono autenticati rispetto a quelli contenuti nel keystore del canale e nell'archivio JRE cacerts .

### **CipherSuite**

CipherSuite è un parametro TLS facoltativo. Viene utilizzato il valore predefinito per provare tutti i CipherSpecabilitati. Deve essere omesso per i canali TCP.

Se si desidera utilizzare una particolare CipherSpec, impostare CipherSuite sul nome della CipherSpec che deve essere utilizzata per stabilire la connessione TLS.

Il servizio di telemetria e il client MQTT negoziano un CipherSpec comune da tutti i CipherSpec abilitati ad ogni estremità. Se una specifica CipherSpec viene specificata in una o in entrambe le estremità della connessione, deve corrispondere alla CipherSpec nell'altra estremità.

Installare ulteriori cifrature aggiungendo ulteriori fornitori a JSSE.

### **FIPS (Federal Information Processing Standards)**

FIPS è un'impostazione facoltativa. Per impostazione predefinita non è impostato.

Nel pannello delle proprietà del gestore code o utilizzando **runmqsc**, impostare SSLFIPS. SSLFIPS specifica se devono essere utilizzati solo algoritmi certificati FIPS.

## Elenco nomi revoche

L'elenco nomi di revoca è un'impostazione facoltativa. Per impostazione predefinita non è impostato.

Nel pannello delle proprietà del gestore code o utilizzando **runmqsc**, impostare SSLCRLNL. SSLCRLNL specifica un elenco nomi degli oggetti delle informazioni di autenticazione utilizzati per fornire le posizioni di revoca dei certificati.

Non viene utilizzato nessun altro parametro del gestore code che imposta le proprietà TLS.

## Client MQTT Java

Impostare le proprietà TLS per il client Java in `MqttConnectionOptions.SSLProperties`; ad esempio:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

I nomi e i valori di specifiche proprietà sono descritti nella classe `MqttConnectOptions`. Per i collegamenti alla documentazione API client per le librerie client MQTT, vedere [Riferimento di programmazione client MQTT](#).

## Protocollo

Protocollo è facoltativo.

Il protocollo viene selezionato in negoziazione con il server di telemetria. Se si richiede un protocollo specifico, è possibile selezionarne uno. Se il server di telemetria non supporta il protocollo, la connessione ha esito negativo.

## ContextProvider

ContextProvider è facoltativo.

## KeyStore

KeyStore è facoltativo. Configurarlo se `ClientAuth` è impostato sul server per forzare l'autenticazione del client.

Inserire il certificato digitale del client, firmato utilizzando la chiave privata, nel keystore. Specificare il percorso e la password del keystore. Il tipo e il fornitore sono facoltativi. JKS è il tipo predefinito e IBMJCE è il fornitore predefinito.

Specificare un provider keystore differente per fare riferimento a una classe che aggiunge un nuovo provider keystore. Inoltre il nome dell'algoritmo utilizzato dal provider keystore per istanziare il `KeyManagerFactory` impostando il nome del gestore chiavi.

## TrustStore

TrustStore è facoltativo. È possibile inserire tutti i certificati attendibili nell'archivio `JRE cacerts`.

Configurare il truststore se si desidera avere un truststore diverso per il client. È possibile che non si configuri il truststore se il server utilizza un certificato emesso da una CA nota che ha già il certificato root memorizzato in `cacerts`.

Aggiungere il certificato firmato pubblicamente del server o il certificato root al truststore e specificare il percorso del truststore e la password. JKS è il tipo predefinito e IBMJCE è il fornitore predefinito.

Specificare un provider truststore differente per fare riferimento a una classe che aggiunge un nuovo provider truststore. Inoltre il nome dell'algoritmo utilizzato dal provider truststore per creare un'istanza di `TrustManagerFactory` impostando il nome del gestore sicuro.

## JRE

Altri aspetti della sicurezza Java che influiscono sul comportamento di TLS sia sul server che sul client sono configurati nel JRE. I file di configurazione su Windows si trovano in *Java Installation Directory\jre\lib\security*. Se si sta utilizzando il JRE fornito con IBM MQ, il percorso è come mostrato nella seguente tabella:

<i>Tabella 18. Percorsi file per piattaforma per file di configurazione TLS JRE</i>	
<b>Piattaforma</b>	<b>Percorso file</b>
Windows	<i>WMQ Installation Directory\java\jre\lib\security</i>
AIX and Linux piattaforme	<i>WMQ Installation Directory/java/jre64/jre/lib/security</i>

### **Autorità di certificazione note**

Il file `cacerts` contiene i certificati root delle autorità di certificazione note. `cacerts` viene utilizzato per impostazione predefinita, a meno che non si specifichi un truststore. Se si utilizza l'archivio `cacerts` o non si fornisce un truststore, è necessario esaminare e modificare l'elenco di firmatari in `cacerts` per soddisfare i propri requisiti di sicurezza.

È possibile aprire `cacerts` utilizzando il comando IBM MQ `strmqikm`, che esegue il programma di utilità IBM Key Management. Aprire `cacerts` come file JKS, utilizzando la password `changeit`. Modificare la password per proteggere il file.

### **Configurazione delle classi di sicurezza**

Utilizzare il file `java.security` per registrare ulteriori provider di sicurezza e altre proprietà di sicurezza predefinite.

### **Autorizzazioni**

Utilizzare il file `java.policy` per modificare le autorizzazioni concesse alle risorse. `javaws.policy` concede le autorizzazioni a `javaws.jar`

### **Livello di crittografia**

Alcuni JRE vengono forniti con una crittografia di potenza ridotta. Se non è possibile importare le chiavi nei keystore, la causa potrebbe essere una codifica di livello ridotto. Provare ad avviare **ikeman** utilizzando il comando **strmqikm** oppure scaricare i file con giurisdizione limitata da [IBM developer kits, Security information](#).

**Importante:** Il tuo paese di origine potrebbe avere restrizioni sull'importazione, il possesso, l'uso o la riesportazione in un altro paese, del software di crittografia. Prima di scaricare o utilizzare i file delle politiche illimitate, è necessario controllare le leggi del proprio Paese. Controllare le relative normative e le relative politiche relative all'importazione, al possesso, all'utilizzo e alla riesportazione del software di crittografia, per determinare se è consentito.

### **Modificare il provider di attendibilità per consentire al client di connettersi a qualsiasi server**

L'esempio illustra come aggiungere un provider sicuro e farvi riferimento dal codice client MQTT. L'esempio non esegue alcuna autenticazione del client o del server. La connessione TLS risultante è codificata senza essere autenticata.

Il frammento di codice in [Figura 19 a pagina 325](#) imposta il provider e il gestore sicuro `AcceptAllProviders` per il client MQTT.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

Figura 19. Frammento di codice MQTT Client

```

package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
private static final long serialVersionUID = 1L;
public AcceptAllProvider() {
super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
put("TrustManagerFactory.TrustAllCertificates",
AcceptAllTrustManagerFactory.class.getName());
}
}

```

Figura 20. AcceptAllProvider.java

```

protected static class AcceptAllTrustManagerFactory extends
javax.net.ssl.TrustManagerFactorySpi {
public AcceptAllTrustManagerFactory() {}
protected void engineInit(java.security.KeyStore keystore) {}
protected void engineInit(
javax.net.ssl.ManagerFactoryParameters parameters) {}
protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
}
}

```

Figura 21. AcceptAllTrustManagerFactory.java

```

protected static class AcceptAllX509TrustManager implements
javax.net.ssl.X509TrustManager {
public void checkClientTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Client authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public void checkServerTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Server authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[0];
}
private static void report(String string) {
System.out.println(string);
}
}
}

```

Figura 22. AcceptAllX509TrustManager.java

Linux

Windows

AIX

## Configurazione del canale di telemetria JAAS

Configurare JAAS per autenticare il Nome utente inviato dal client.

L'amministratore IBM MQ configura quali canali MQTT richiedono l'autenticazione client utilizzando JAAS. Specificare il nome di una configurazione JAAS per ogni canale che deve eseguire l'autenticazione JAAS. I canali possono utilizzare la stessa configurazione JAAS oppure possono utilizzare configurazioni JAAS differenti. Le configurazioni sono definite in *WMQData directory\qmgrs\qMgrName\mqxr\jaas.config*.

Il file *jaas.config* è organizzato per nome di configurazione JAAS. Sotto ogni nome di configurazione si trova un elenco di configurazioni di login; consultare ["File jaas.config di esempio"](#) a pagina 327.

JAAS fornisce quattro moduli di login standard. I moduli NT standard e UNIX Login hanno un valore limitato.

## Modulo JndiLogin

Esegue l'autenticazione rispetto a un servizio directory configurato in JNDI ( Java Naming and Directory Interface).

## Krb5LoginModule

Autentica utilizzando i protocolli Kerberos .

## NTLoginModule

Autentica utilizzando le informazioni di sicurezza NT per l'utente corrente.

## Modulo UnixLogin

Autentica utilizzando le informazioni di sicurezza UNIX per l'utente corrente.

Il problema nell'utilizzo di NTLoginModule o UnixLoginModule è che il servizio di telemetria (MQXR) viene eseguito con l'identità mqm e non con l'identità del canale MQTT . mqm è l'identità passata a NTLoginModule o UnixLoginModule per l'autenticazione e non l'identità del client.

Per risolvere questo problema, scrivere il proprio modulo di login o utilizzare gli altri moduli di login standard. Un esempio di JAASLoginModule . java viene fornito con MQ Telemetry. È un'implementazione dell'interfaccia javax . security . auth . spi . LoginModule . Utilizzarla per sviluppare il proprio metodo di autenticazione.

Tutte le nuove classi LoginModule fornite devono trovarsi nel percorso classi del servizio di telemetria (MQXR). Non inserire le classi nelle directory IBM MQ che si trovano nel percorso di classe. Creare le proprie directory e definire l'intero percorso classe per il servizio di telemetria (MQXR).

È possibile incrementare il percorso classe utilizzato dal servizio di telemetria (MQXR) impostando il percorso classe nel file service . env . CLASSPATH deve essere in maiuscolo e l'istruzione del percorso classe può contenere solo valori letterali. Non è possibile utilizzare variabili in CLASSPATH; ad esempio, CLASSPATH=%CLASSPATH% non è corretto. Il servizio di telemetria (MQXR) imposta il proprio percorso classi. Il CLASSPATH definito in service . env viene aggiunto ad esso.

Il servizio di telemetria (MQXR) fornisce due callback che restituiscono il Nome utente e la Password per un client connesso al canale MQTT . Nome utente e Password sono impostati nell'oggetto MqttConnectOptions . Consultare [“Metodo JAASLoginModule.Login\(\) di esempio” a pagina 327](#) per un esempio di come accedere a Nome utente e Password.

## File jaas . config di esempio

Un esempio di file di configurazione JAAS con una configurazione denominata, MQXRConfig

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//      principal=principal@your_realm
//      useDefaultCcache=TRUE
//      renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//      useTicketCache="true"
//      ticketCache="$${user.home}${/}tickets";
};
```

## Metodo JAASLoginModule.Login() di esempio

Un esempio di modulo di accesso JAAS codificato per ricevere il Nome utente e la Password forniti da un client MQTT .

```
public boolean login()
throws javax.security.auth.login.LoginException {
javax.security.auth.callback.Callback[] callbacks =
new javax.security.auth.callback.Callback[2];
callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
callbacks[1] = new javax.security.auth.callback.PasswordCallback(
"PasswordCallback", false);
```

```

try {
callbackHandler.handle(callbacks);
String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
.getName();
char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
.getPassword();
// Accept everything.
if (true) {
loggedIn = true;
} else
throw new javax.security.auth.login.FailedLoginException("Login failed");

principal= new JAASPrincipal(username);

} catch (java.io.IOException exception) {
throw new javax.security.auth.login.LoginException(exception.toString());
} catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
throw new javax.security.auth.login.LoginException(exception.toString());
}
}

return loggedIn;
}

```

### Attività correlate

Risoluzione del problema: [modulo di login di JAAS non richiamato dal servizio di telemetria](#)

### Riferimenti correlati

[Classe MQXR AuthCallback](#)

## Gestione di un client AMQP

È possibile amministrare un client AMQP utilizzando IBM MQ Explorer o su una riga comandi. Utilizzare Explorer per configurare canali e monitorare client AMQP connessi a IBM MQ. Configurare la protezione dei client AMQP utilizzando TLS e JAAS.

### Prima di iniziare

Per informazioni sull'installazione di AMQP sulla tua piattaforma, vedi [Scelta di cosa installare](#).

### Amministrazione mediante IBM MQ Explorer

Utilizzare Explorer per configurare canali AMQP e monitorare i client AMQP connessi a IBM MQ. È possibile configurare la sicurezza dei client AMQP utilizzando TLS e JAAS.

### Gestione mediante la riga comandi

È possibile gestire un client AMQP dalla riga comandi [utilizzando i comandi MQSC](#).

## Visualizzazione degli oggetti IBM MQ utilizzati dai clienti AMQP

È possibile visualizzare le diverse risorse IBM MQ utilizzate dai client AMQP, ad esempio le connessioni e le sottoscrizioni.

### Connessioni

Quando il servizio AMQP viene avviato, vengono create nuove Hconn e connesse al gestore code. Questo pool di Hconn viene utilizzato quando i client AMQP pubblicano i messaggi. È possibile visualizzare le Hconn utilizzando il comando **DISPLAY CONN**. Ad esempio:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM MQ Advanced Message Queuing Protocol*')
```

Questo comando mostra anche tutte le Hconn specifiche del client. Le Hconn che dispongono di un attributo ID client vuoto sono le Hconn utilizzate nel pool

Quando un client AMQP si connette a un canale AMQP, un nuovo Hconn è connesso al gestore code. Questo Hconn viene utilizzato per utilizzare i messaggi in maniera asincrona per le sottoscrizioni che il

client AMQP ha creato. È possibile visualizzare l'Hconn utilizzato da un particolare client AMQP utilizzando il comando **DISPLAY CONN**. Ad esempio:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

## Sottoscrizioni create dai client

Quando un client AMQP sottoscrive un topic, viene creata una nuova sottoscrizione IBM MQ. Il nome della sottoscrizione contiene le seguenti informazioni:

- Il nome del cliente. Se il client si è unito a una sottoscrizione condivisa, viene utilizzato il nome della condivisione
- Il modello di argomento a cui il client ha eseguito la sottoscrizione
- Un prefisso. Il prefisso è `private` se il client ha creato una sottoscrizione non condivisa o `share` se il client si è unito a una sottoscrizione condivisa

Per visualizzare le sottoscrizioni utilizzate da un particolare client AMQP, eseguire il comando **DISPLAY SUB** e filtrare in base al prefisso `private`:

```
DISPLAY SUB('/:private:*')
```

Per visualizzare le sottoscrizioni condivise utilizzate da più client, eseguire il comando **DISPLAY SUB** e filtrare in base al prefisso `share`:

```
DISPLAY SUB('/:share:*')
```

Poiché le sottoscrizioni condivise possono essere utilizzate da più client AMQP, è possibile visualizzare i client che attualmente utilizzano i messaggi dalla sottoscrizione condivisa. Puoi farlo elencando le Hconn che attualmente hanno un handle aperto nella coda di sottoscrizione. Per visualizzare i client che attualmente utilizzano una condivisione, completare la seguente procedura:

1. Trovare il nome della coda che la sottoscrizione condivisa utilizza come destinazione. Ad esempio:

```
DISPLAY SUB('/:private:recv_e298452:public') DEST
5 : DISPLAY SUB('/:private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D312020202020202020202020707E0A565C2D0020)
SUB('/:private:recv_e298452:public')
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
```

2. Eseguire il comando **DISPLAY CONN** per trovare gli handle aperti su tale coda:

```
DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME
EQ SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
21 : DISPLAY CONN(*) TYPE(HANDLE) WHERE(OBJNAME EQ
SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(HANDLE)

OBJNAME(SYSTEM.BASE.TOPIC)          OBJTYPE(TOPIC)

OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(QUEUE)
```

3. Per ogni handle, visualizzare l'ID del client AMQP con l'handle aperto:

```
DISPLAY CONN(707E0A56642B0020) CLIENTID
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
```

```
CONN(707E0A56642B0020)
EXTCONN(414D5143514D312020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
24 : DISPLAY CONN(707E0A565F290020) CLIENTID
AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D312020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)
```

## Identificazione, autorizzazione e autenticazione del client AMQP

Come altre applicazioni client IBM MQ , è possibile proteggere le connessioni AMQP in diversi modi.

È possibile utilizzare le seguenti funzioni di protezione per proteggere le connessioni AMQP a IBM MQ:

- [Record di autenticazione di canale](#)
- [Autenticazione connessione](#)
- Configurazione utente MCA canale
- Definizioni di autorizzazione IBM MQ
- [Connettività TLS](#)

Da una prospettiva di sicurezza, stabilire una connessione consiste nei seguenti due passaggi:

- Decidere se la connessione deve continuare
- Decisione dell'identità IBM MQ che l'applicazione assume per i successivi controlli di autorizzazione

Le seguenti informazioni illustrano le diverse configurazioni di IBM MQ e i passi che vengono eseguiti quando un client AMQP tenta di stabilire una connessione. Non tutte le configurazioni IBM MQ utilizzano tutte le operazioni descritte. Ad esempio, alcune configurazioni non utilizzano TLS per le connessioni all'interno del firewall aziendale e alcune configurazioni utilizzano TLS ma non utilizzano i certificati client per l'autenticazione. Molti ambienti non utilizzano moduli JAAS personalizzati o personalizzati.

### Stabilire una connessione

La seguente procedura descrive cosa accade quando viene stabilita una connessione da un client AMQP. La procedura determina se la connessione continua e quale identità IBM MQ assume l'applicazione per i controlli di autorizzazione:

1. Se il client apre una connessione TLS a IBM MQ e fornisce un certificato, il gestore code tenta di convalidarlo.
2. Se il client fornisce credenziali nome utente e password, viene ricevuto un frame SASL AMQP dal gestore code e viene controllata la configurazione CONNAUTH di MQ .
3. Le regole di autenticazione del canale MQ vengono controllate (ad esempio, se l'indirizzo IP e il DN del certificato TLS sono validi)
4. Il canale MCAUSER viene asserito, a meno che le regole di autenticazione del canale non determinino diversamente.
5. Se è stato configurato un modulo JAAS , viene richiamato
6. MQ Controllo autorizzazione CONNECT applicato all'ID utente MQ risultante.
7. Connessione stabilita con un'identità IBM MQ assunta.

### Pubblicazione di un messaggio

La seguente procedura descrive cosa accade quando un messaggio viene pubblicato da un client AMQP. La procedura determina se la connessione continua e quale identità IBM MQ assume l'applicazione per i controlli di autorizzazione:

1. Il frame di collegamento AMQP arriva al gestore code. L'autorizzazione di pubblicazione IBM MQ per la stringa di argomenti specificata viene controllata per l'identità utente MQ stabilita durante la connessione.
2. Il messaggio viene pubblicato nella stringa argomento specificata.

## Sottoscrizione a un pattern di argomento

La seguente procedura descrive cosa accade quando un client AMQP sottoscrive un pattern di argomento. La procedura determina se la connessione continua e quale identità IBM MQ assume l'applicazione per i controlli di autorizzazione:

1. Il frame di collegamento AMQP arriva al gestore code. L'autorizzazione di sottoscrizione IBM MQ per il modello di argomento specificato viene controllata per l'identit ... utente MQ stabilita durante la connessione.
2. La sottoscrizione è stata creata.

## Identità e autorizzazione del client AMQP

Utilizzare l'ID client AMQP, il nome utente AMQP o un'identità client comune definita sul canale o in una regola di autenticazione del canale, per l'autorizzazione ad accedere agli oggetti IBM MQ .

L'amministratore effettua la scelta quando definisce o modifica il canale AMQP, configurando l'impostazione CONNAUTH del gestore code o definendo le regole di autenticazione del canale. L'identità viene utilizzata per autorizzare l'accesso agli argomenti IBM MQ . La scelta viene effettuata in base a quanto segue:

1. L'attributo USECLNTID del canale.
2. L'attributo ADOPTCTX della regola CONNAUTH del gestore code.
3. L'attributo MCAUSER definito nel canale.
4. L'attributo USERSRC di una regola di autenticazione di canale corrispondente.

**Prevenzione dei problemi:** L'identità scelta da questo processo viene successivamente indicata, ad esempio dal comando AMQP (DISPLAY CHSTATUS), come MCAUSER del client. Tenere presente che questa non è necessariamente la stessa identità del MCAUSER del canale a cui si fa riferimento nella scelta (2).

Utilizzare il comando IBM MQ **setmqaut** per selezionare quali oggetti e quali azioni sono autorizzati ad essere utilizzati dall'identit ... associata al canale AMQP. Ad esempio, i seguenti comandi autorizzano un'identità del canale AMQPClient, fornita dall'amministratore del gestore code QM1:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPClient -all +pub +sub
```

e

```
setmqaut -m QM1 -t qmgr -p AMQPClient -all +connect
```

## Autenticazione client AMQP utilizzando una password

Autenticare il nome utente del client AMQP utilizzando la password del client. È possibile autenticare il client utilizzando un'identità diversa da quella utilizzata per autorizzare il client a pubblicare e sottoscrivere argomenti.

Il servizio AMQP può utilizzare MQ CONNAUTH o JAAS per autenticare il nome utente del client. Se uno di questi è configurato, la password fornita dal client viene verificata dalla configurazione CONNAUTH di MQ o dal modulo JAAS .

La seguente procedura descrive i passi di esempio per autenticare i singoli utenti rispetto agli utenti e alle password del sistema operativo locale e, in caso di esito positivo, adottare l'identità comune AMQPUser:

1. L'amministratore di IBM MQ imposta l'identità MCAUSER del canale AMQP su qualsiasi nome, ad esempio AMQPUser, utilizzando Esplora risorse di IBM MQ.
2. L'amministratore IBM MQ autorizza AMQPUser a pubblicare e sottoscrivere qualsiasi argomento:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. L'amministratore IBM MQ configura una regola IDPWOS CONNAUTH per controllare il nome utente e password presentati dal client. La regola CONNAUTH deve impostare CHCKCLNT (REQUIRED) e ADOPTCTX (NO).

**Nota:** Si consiglia di utilizzare le regole di autenticazione di canale e di impostare l'attributo del canale MCAUSER su un utente che non dispone di privilegi, per consentire un maggiore controllo sulle connessioni al gestore code.

## Privacy delle pubblicazioni sui canali

La privacy delle pubblicazioni AMQP inviate in entrambe le direzioni attraverso i canali AMQP è protetta utilizzando TLS per codificare le trasmissioni sulla connessione.

I client AMQP che si collegano ai canali AMQP utilizzano TLS per proteggere la privacy delle pubblicazioni trasmesse sul canale utilizzando la crittografia a chiave simmetrica. Poiché gli endpoint non sono autenticati, non puoi considerare attendibile la sola crittografia del canale. Combina la protezione della privacy con il server o l'autenticazione reciproca.

Come alternativa all'utilizzo di TLS, alcuni tipi di VPN (Virtual Private Network), come IPsec, autenticano gli endpoint di una connessione TCP/IP. VPN codifica ogni pacchetto IP che passa attraverso la rete. Una volta che tale connessione VPN è stabilita, è stata creata una rete attendibile. È possibile connettere i client AMQP ai canali AMQP utilizzando TCP/IP sulla rete VPN.

La crittografia delle connessioni TLS senza autenticare il server espone la connessione agli attacchi man-in-the-middle. Anche se le informazioni scambiate sono protette contro le intercettazioni, non sai con chi le stai scambiando. A meno che tu non controlli la rete, sei esposto a qualcuno che intercetta le tue trasmissioni IP e si maschera come l'endpoint.

Puoi creare una connessione TLS crittografata, senza autenticare il server, utilizzando un CipherSpec di scambio di chiavi Diffie - Hellman che supporta TLS anonimo. Il segreto master, condiviso tra il client e il server e utilizzato per codificare le trasmissioni TLS, viene stabilito senza scambiare un certificato server firmato privatamente.

Poiché le connessioni anonime non sono sicure, la maggior parte delle implementazioni TLS non utilizzano per impostazione predefinita CipherSpec anonime. Se una richiesta client per la connessione TLS viene accettata da un canale AMQP, il canale deve avere un keystore protetto da una passphrase. Per impostazione predefinita, poiché le implementazioni TLS non utilizzano CipherSpec anonime, il keystore deve contenere un certificato firmato privatamente che il client può autenticare.

Se si utilizzano CipherSpec anonimi, il keystore del server deve esistere, ma non deve contenere alcun certificato firmato privatamente.

Un altro modo per stabilire una connessione crittografata consiste nel sostituire il provider di attendibilità sul client con la propria implementazione. Il provider di attendibilità non autentica il certificato del server, ma la connessione viene codificata.

## Configurazione dei client AMQP con TLS

È possibile configurare i client AMQP per utilizzare TLS per proteggere i dati in transito nella rete e per autenticare l'identità del gestore code a cui si connette il client.

Per utilizzare TLS per la connessione da un client AMQP a un canale AMQP, è necessario assicurarsi che il gestore code sia stato configurato per TLS. [Configurazione di TLS sui gestori code](#) descrive come configurare il keystore da cui un gestore code legge i certificati TLS.

Quando il gestore code è stato configurato con un keystore, è necessario configurare gli attributi TLS sul canale AMQP a cui si conetteranno i client. I canali AMQP hanno quattro attributi correlati alla configurazione TLS come segue:

#### **SSLCAUTH**

L'attributo SSLCAUTH viene utilizzato per specificare se il gestore code deve richiedere un client AMQP per presentare un certificato client per verificarne l'identità.

#### **SSLCIPH**

L'attributo SSLCIPH specifica la codifica che il canale deve utilizzare per codificare i dati nel flusso TLS.

#### **SSLPEER**

L'attributo SSLPEER viene utilizzato per specificare il DN (distinguished name) a cui deve corrispondere un certificato client se deve essere consentita una connessione.

#### **CERTLABL**

CERTLABL specifica il certificato che il gestore code deve presentare al client. Il keystore del gestore code può contenere più certificati. Questo attributo consente di specificare il certificato da utilizzare per le connessioni a questo canale. Se non viene specificato alcun CERTLABL, viene utilizzato il certificato nel contenitore di chiavi del gestore code con l'etichetta corrispondente all'attributo CERTLABL del gestore code.

Una volta configurato il canale AMQP con gli attributi TLS, è necessario riavviare il servizio AMQP utilizzando il seguente comando:

```
STOP SERVICE(SYSTEM.AMQP.SERVICE) START SERVICE(SYSTEM.AMQP.SERVICE)
```

Quando un client AMQP si connette a un canale AMQP protetto da TLS, il client verifica l'identità del certificato presentato dal gestore code. A tale scopo, è necessario configurare il proprio client AMQP con un truststore contenente il certificato del gestore code. I passi per eseguire questa operazione variano a seconda del client AMQP che si sta utilizzando. Per informazioni sui vari client e API AMQP, consultare la rispettiva documentazione del client AMQP.

## **Disconnessione dei client AMQP dal gestore code**

Se si desidera disconnettere i client AMQP dal gestore code, eseguire il comando `PURGE CHANNEL` oppure arrestare la connessione al client AMQP.

- Eseguire il comando **`PURGE CHANNEL`** . Ad esempio:

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- In alternativa, arrestare la connessione che il client AMQP sta utilizzando per disconnettere il client completando la seguente procedura:

1. Individuare la connessione utilizzata dal client eseguendo il comando **`DISPLAY CONN`** . Ad esempio:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

L'output del comando è il seguente:

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_28dbb7e)
```

2. Arrestare la connessione. Ad esempio:

## Gestione multicast

Utilizzare queste informazioni per informazioni sulle attività di gestione di IBM MQ Multicast, come la riduzione della dimensione dei messaggi multicast e l'abilitazione della conversione dei dati.

### Introduzione a multicast

Utilizzare queste informazioni per iniziare a utilizzare gli argomenti IBM MQ Multicast e gli oggetti delle informazioni di comunicazione.

#### Informazioni su questa attività

IBM MQ La messaggistica multicast utilizza la rete per consegnare i messaggi associando gli argomenti agli indirizzi di gruppo. Le seguenti attività sono un modo rapido per verificare se l'indirizzo IP e la porta richiesti sono configurati correttamente per la messaggistica multicast.

#### Creazione di un oggetto COMMINFO per multicast

L'oggetto informazioni di comunicazione (COMMINFO) contiene gli attributi associati alla trasmissione multicast. Per ulteriori informazioni sui parametri oggetto COMMINFO, consultare [DEFINE COMMINFO](#).

Utilizzare il seguente esempio di riga comandi per definire un oggetto COMMINFO per multicast:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

dove *MC1* è il nome dell'oggetto COMMINFO, *indirizzo gruppo* è l'indirizzo IP o il nome DNS multicast del gruppo e il *numero porta* è la porta su cui trasmettere (il valore predefinito è 1414).

Viene creato un nuovo oggetto COMMINFO denominato *MC1* ; questo nome è il nome che è necessario specificare quando si definisce un oggetto TOPIC nell'esempio successivo.

#### Creazione di un oggetto TOPIC per multicast

Un argomento è l'oggetto delle informazioni pubblicate in un messaggio di pubblicazione / sottoscrizione e un argomento viene definito creando un oggetto TOPIC. Gli oggetti TOPIC hanno due parametri che definiscono se possono essere utilizzati o meno con multicast. Questi parametri sono **COMMINFO** e **MCAST**.

- **COMMINFO** Questo parametro specifica il nome dell'oggetto delle informazioni di comunicazione multicast. Per ulteriori informazioni sui parametri oggetto COMMINFO, consultare [DEFINE COMMINFO](#).
- **MCAST** Questo parametro specifica se il multicast è consentito in questa posizione nella struttura ad albero degli argomenti.

Utilizzare il seguente esempio di riga comandi per definire un oggetto TOPIC per multicast:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

Viene creato un nuovo oggetto TOPIC denominato *ALLSPORTS* . Ha una stringa di argomenti *Sports*, il relativo oggetto delle informazioni di comunicazione è denominato *MC1* (che è il nome specificato durante la definizione di un oggetto COMMINFO nell'esempio precedente) e multicast è abilitato.

#### Verifica della pubblicazione / sottoscrizione multicast

Dopo la creazione degli oggetti TOPIC e COMMINFO, è possibile verificarli utilizzando l'esempio `amqspubc` e l'esempio `amqssubc` . Per ulteriori informazioni su questi esempi, consultare [Programmi di esempio di pubblicazione / sottoscrizione](#).

1. Aprire due finestre della riga comandi; la prima riga comandi è per l'esempio di pubblicazione `amqspubc` e la seconda riga comandi è per l'esempio di sottoscrizione `amqssubc`.
2. Immettere il seguente comando nella riga comandi 1:

```
amqspubc Sports QM1
```

dove *Sports* è la stringa di argomenti dell'oggetto TOPIC definito in un esempio precedente e *QM1* è il nome del gestore code.

3. Immettere il seguente comando nella riga comandi 2:

```
amqssubc Sports QM1
```

dove *Sports* e *QM1* sono gli stessi utilizzati nel passaggio “2” a pagina 335.

4. Immettere `Hello world` alla riga comandi 1. Se la porta e l'indirizzo IP specificati nell'oggetto `COMMINFO` sono configurati correttamente; l'esempio `amqssubc`, che è in ascolto sulla porta per le pubblicazioni dall'indirizzo specificato, emette `Hello world` sulla riga comandi 2.

## Topologia argomento IBM MQ Multicast

Utilizzare questo esempio per comprendere la topologia dell'argomento IBM MQ Multicast.

IBM MQ Il supporto multicast richiede che ogni struttura ad albero secondaria abbia il proprio gruppo multicast e flusso di dati all'interno della gerarchia totale.

Lo schema di reindirizzamento IP *rete classful* ha uno spazio degli indirizzi designato per l'indirizzo multicast. L'intervallo multicast completo dell'indirizzo IP è compreso tra 224.0.0.0 e 239.255.255.255 ma alcuni di questi indirizzi sono riservati. Per un elenco di indirizzi riservati, contattare l'amministratore del sistema oppure visualizzare <https://www.iana.org/assignments/multicast-addresses> per ulteriori informazioni. Si consiglia di utilizzare l'indirizzo multicast di ambito locale nell'intervallo compreso tra 239.0.0.0 e 239.255.255.255.

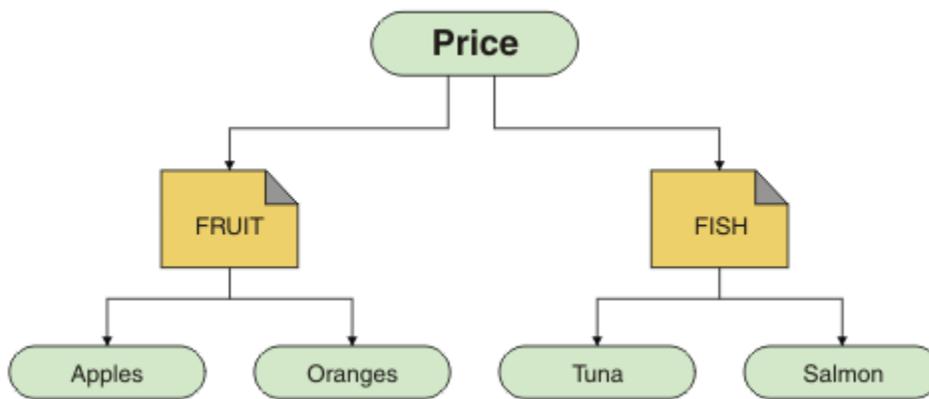
Nel seguente diagramma, sono disponibili due possibili flussi di dati multicast:

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX
)
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

dove 239.XXX.XXX.XXX e 239.YYY.YYY.YYY sono indirizzi multicast validi.

Queste definizioni di argomenti vengono utilizzate per creare una struttura ad albero degli argomenti come mostrato nel seguente diagramma:

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



Ogni oggetto informazioni di comunicazione multicast (COMMINFO) rappresenta un flusso di dati differente poiché i rispettivi indirizzi di gruppo sono diversi. In questo esempio, l'argomento FRUIT è definito per utilizzare l'oggetto COMMINFO MC1, l'argomento FISH è definito per utilizzare l'oggetto COMMINFO MC2 e il nodo Price non dispone di definizioni multicast.

IBM MQ Il multicast ha un limite di 255 caratteri per le stringhe di argomento. Questa limitazione indica che è necessario prestare attenzione con i nomi dei nodi e dei nodi foglia all'interno della struttura ad albero; se i nomi dei nodi e dei nodi foglia sono troppo lunghi, la stringa dell'argomento potrebbe superare 255 caratteri e restituire il codice motivo 2425 (0979) (RC2425): MQRC\_TOPIC\_STRING\_ERROR. Si consiglia di rendere le stringhe di argomenti il più brevi possibile perché le stringhe di argomenti più lunghe potrebbero avere un effetto negativo sulle prestazioni.

## Controllo della dimensione dei messaggi multicast

Utilizzare queste informazioni per informazioni sul formato dei messaggi IBM MQ e ridurre la dimensione dei messaggi IBM MQ.

I messaggi IBM MQ hanno un numero di attributi associati che sono contenuti nel descrittore del messaggio. Per messaggi di piccole dimensioni, questi attributi potrebbero rappresentare la maggior parte del traffico dati e possono avere un effetto negativo significativo sulla velocità di trasmissione. IBM MQ Multicast consente all'utente di configurare quali, se presenti, di questi attributi vengono trasmessi insieme al messaggio.

La presenza di attributi del messaggio, diversi dalla stringa dell'argomento, dipende dal fatto che l'oggetto COMMINFO indichi che devono essere inviati o meno. Se un attributo non viene trasmesso, l'applicazione ricevente applica un valore predefinito. I valori MQMD predefiniti non corrispondono necessariamente al valore MQMD\_DEFAULT e sono descritti in Tabella 19 a pagina 337.

L'oggetto COMMINFO contiene l'attributo MCPROP che controlla il numero di campi MQMD e il flusso di proprietà utente con il messaggio. Impostando il valore di questo attributo su un livello appropriato, è possibile controllare la dimensione dei messaggi multicast IBM MQ:

### MCPROP

Il controllo proprietà multicast verifica quante proprietà utente e MQMD vengono trasmesse insieme al messaggio.

### TUTTO

Vengono trasmesse tutte le proprietà utente e tutti i campi di MQMD.

### Rispondi

Solo le proprietà utente e i campi MQMD che si occupano delle risposte ai messaggi vengono trasmessi. Queste proprietà sono:

- MsgType
- MessageId
- CorrelId
- ReplyToQ

- ReplyToQmgr

#### USER

Solo le proprietà utente vengono trasmesse.

#### NESSUNO

Non vengono trasmessi né le proprietà utente, né i campi MQMD.

#### COMPAT

Questo valore fa in modo che la trasmissione del messaggio venga eseguita in modalità compatibile a RMM, il che consente alcune interoperazioni con le applicazioni XMS correnti e IBM Integration Bus RMM .

 XMS .NET La messaggistica multicast (utilizzando RMM) è obsoleta da IBM MQ 9.2 e verrà rimossa in una release futura di XMS .NET.

## Attributi messaggio multicast

Gli attributi dei messaggi possono provenire da varie posizioni, come MQMD, i campi in MQRFH2e le proprietà dei messaggi.

La tabella riportata di seguito mostra cosa accade quando i messaggi vengono inviati in base al valore di MCPROP (descritto in precedenza in questa sezione) e il valore predefinito utilizzato quando un attributo non viene inviato.

<i>Tabella 19. Attributi di messaggistica e modalità di relazione con multicast</i>		
<b>Attributo</b>	<b>Azione quando si utilizza multicast</b>	<b>Valore predefinito se non trasmesso</b>
TopicString	Sempre incluso	Non applicabile
StrucId MQMQ	Non trasmesso	Non applicabile
Versione MQMD	Non trasmesso	Non applicabile
Prospetto	Incluso se non predefinito	0
MsgType	Incluso se non predefinito	MQMT_DATAGRAM
Scadenza	Incluso se non predefinito	0
Feedback	Incluso se non predefinito	0
Codifica	Incluso se non predefinito	MQENC_NORMAL (equivalente)
CodedCharSetId	Incluso se non predefinito	1208
Formato	Incluso se non predefinito	MQRFH2
Priorit...	Incluso se non predefinito	4
Persistenza	Incluso se non predefinito	MQPER_NOT_PERSISTENT
MsgId	Incluso se non predefinito	Nulla
CorrelId	Incluso se non predefinito	Nulla
BackoutCount	Incluso se non predefinito	0
ReplyToQ	Incluso se non predefinito	Spazio
ReplyToQMgr	Incluso se non predefinito	Spazio
UserIdentifier	Incluso se non predefinito	Spazio
AccountingToken	Incluso se non predefinito	Nulla
Tipo IT PutApp	Incluso se non predefinito	JAVA MQAT

Tabella 19. Attributi di messaggistica e modalità di relazione con multicast (Continua)

Attributo	Azione quando si utilizza multicast	Valore predefinito se non trasmesso
PutAppIName	Incluso se non predefinito	Spazio
PutDate	Incluso se non predefinito	Spazio
PutTime	Incluso se non predefinito	Spazio
ApplOriginData	Incluso se non predefinito	Spazio
GroupID	Esclusi	Non applicabile
MsgSeqNumber	Esclusi	Non applicabile
Offset	Esclusi	Non applicabile
MsgFlags	Esclusi	Non applicabile
OriginalLength	Esclusi	Non applicabile
UserProperties	Inclusi	Non applicabile

#### Riferimenti correlati

[COMMINFO ALTER](#)

[DEFINE COMMINFO](#)

## Abilitazione della conversione dati per la messaggistica multicast

Utilizzare queste informazioni per comprendere il funzionamento della conversione dei dati per la messaggistica IBM MQ Multicast.

IBM MQ Multicast è un protocollo condiviso, senza connessione e quindi non è possibile per ogni client effettuare richieste specifiche per la conversione dei dati. Ogni client sottoscritto allo stesso flusso multicast riceve gli stessi dati binari; pertanto, se è richiesta la conversione dei dati IBM MQ, la conversione viene eseguita localmente su ciascun client.

In un'installazione a piattaforma mista, è possibile che la maggior parte dei client richieda i dati in un formato che non sia il formato nativo dell'applicazione di trasmissione. In questa situazione, i valori **CCSID** e **ENCODING** dell'oggetto COMMINFO multicast possono essere utilizzati per definire la codifica della trasmissione del messaggio per una maggiore efficienza.

IBM MQ Multicast supporta la conversione dei dati del payload del messaggio per i seguenti formati integrati:

- MQADMIN
- MQEVENT
- MQPCF
- MQRFH
- MQRFH2
- MQSTR

Oltre a questi formati, è anche possibile definire i propri formati e utilizzare un'exit di conversione dati [MQDXP](#) - parametro di conversione dati .

Per informazioni sulla programmazione delle conversione dei dati, consultare [Conversione dei dati in MQI per la messaggistica multicast](#).

Per ulteriori informazioni sulla conversione dati, consultare [Conversione dati](#).

Per ulteriori informazioni sulle uscite di conversione dati e `ClientExitPath`, consultare la sezione [ClientExitPercorso](#) del file di configurazione client.

## Monitoraggio applicazione multicast

Utilizzare queste informazioni per informazioni sulla gestione e il monitoraggio di IBM MQ Multicast.

Lo stato dei publisher e dei sottoscrittori correnti per il traffico multicast (ad esempio, il numero di messaggi inviati e ricevuti o il numero di messaggi persi) viene periodicamente trasmesso al server dal client. Quando si riceve lo stato, l'attributo **COMMEV** dell'oggetto **COMMINFO** specifica se il gestore code inserisce o meno un messaggio di evento nel **SISTEMA.SYSTEM.ADMIN.PUBSUB.EVENT**. Il messaggio dell'evento contiene le informazioni sullo stato ricevute. Queste informazioni sono un aiuto diagnostico inestimabile per individuare l'origine di un problema.

Utilizzare il comando MQSC **DISPLAY CONN** per visualizzare le informazioni di connessione relative alle applicazioni connesse al gestore code. Per ulteriori informazioni relative al comando **DISPLAY CONN**, consultare [DISPLAY CONN](#).

Utilizzare il comando MQSC **DISPLAY TPSTATUS** per visualizzare lo stato dei publisher e dei sottoscrittori. Per ulteriori informazioni relative al comando **DISPLAY TPSTATUS**, consultare [DISPLAY TPSTATUS](#).

### COMMEV e l'indicatore di affidabilità del messaggio multicast

L'*indicatore di affidabilità*, utilizzato insieme all'attributo **COMMEV** dell'oggetto **COMMINFO**, è un elemento chiave nel monitoraggio dei publisher e dei sottoscrittori IBM MQ Multicast. L'indicatore di affidabilità (il campo **MSGREL** che viene restituito nei comandi di stato di pubblicazione o sottoscrizione) è un indicatore IBM MQ che illustra la percentuale di trasmissioni che non hanno errori. A volte i messaggi devono essere ritrasmessi a causa di un errore di trasmissione, che si riflette nel valore di **MSGREL**. Le cause potenziali degli errori di trasmissione includono i sottoscrittori lenti, le reti occupate e le interruzioni di rete. **COMMEV** controlla se i messaggi di eventi vengono generati per handle multicast creati utilizzando l'oggetto **COMMINFO** ed è impostato su uno dei tre valori possibili:

#### DISABILITATO

I messaggi di evento non vengono scritti.

#### Abilitato

I messaggi di evento vengono sempre scritti, con una frequenza definita nel parametro **COMMINFO MONINT**.

#### ECCEZIONE

I messaggi di evento vengono scritti se l'affidabilità del messaggio è inferiore alla soglia di affidabilità. Un livello di affidabilità del messaggio pari o inferiore al 90% indica che potrebbe essersi verificato un problema con la configurazione della rete o che una o più applicazioni di pubblicazione / sottoscrizione sono in esecuzione troppo lentamente:

- Un valore **MSGREL (100, 100)** indica che non si sono verificati problemi nel breve termine o nel periodo di tempo a lungo termine.
- Il valore **MSGREL (80, 60)** indica che il 20% dei messaggi ha attualmente problemi, ma rappresenta anche un miglioramento rispetto al valore a lungo termine di 60.

I client potrebbero continuare a trasmettere e ricevere traffico multicast anche quando la connessione unicast al gestore code è interrotta, pertanto i dati potrebbero non essere aggiornati.

## Affidabilità dei messaggi multicast

Utilizzare queste informazioni per informazioni su come impostare la sottoscrizione multicast IBM MQ e la cronologia dei messaggi.

Un elemento chiave per superare l'errore di trasmissione con multicast è il buffer dei dati trasmessi (una cronologia dei messaggi che devono essere conservati all'estremità di trasmissione del link) da IBM MQ. Questo processo significa che non è richiesta alcuna memorizzazione nel buffer dei messaggi nel processo di inserimento dell'applicazione poiché IBM MQ fornisce l'affidabilità. La dimensione di questa cronologia viene configurata mediante l'oggetto informazioni di comunicazione (**COMMINFO**), come descritto nelle seguenti informazioni. Un buffer di trasmissione più grande significa che c'è più

cronologia di trasmissione da ritrasmettere se necessario, ma a causa della natura del multicast, la consegna garantita al 100% non può essere supportata.

La cronologia dei messaggi multicast IBM MQ è controllata nell'oggetto delle informazioni di comunicazione (COMMINFO) dall'attributo **MSGHIST** :

#### **MSGHIST**

Questo valore è la quantità di cronologia dei messaggi in kilobyte conservata dal sistema per gestire le ritrasmissioni nel caso di NACK (riconoscimenti negativi).

Il valore 0 fornisce il livello minimo di affidabilità. Il valore predefinito è 100 KB.

La cronologia della nuova sottoscrizione IBM MQ Multicast è controllata nell'oggetto informazioni di comunicazione (COMMINFO) dall'attributo **NSUBHIST** :

#### **NSUBHIST**

La cronologia nuovo sottoscrittore verifica se un sottoscrittore che si iscrive a un flusso di pubblicazioni riceve tutti dati attualmente disponibili o solo le pubblicazioni disponibili dal momento della sottoscrizione.

#### **NESSUNO**

Un valore di NONE fa sì che il trasmettitore trasmetta solo la pubblicazione effettuata dal momento della sottoscrizione. NONE è il valore predefinito.

#### **TUTTO**

Un valore ALL fa sì che il trasmettitore ritrasmetta la quantità di cronologia dell'argomento nota. In alcune circostanze, questa situazione può fornire un comportamento simile alle pubblicazioni conservate.

**Nota:** L'utilizzo del valore di ALL potrebbe avere un effetto negativo sulle prestazioni se esiste una cronologia di argomenti di grandi dimensioni poiché tutta la cronologia degli argomenti viene ritrasmessa.

#### **Riferimenti correlati**

[DEFINE COMMINFO](#)

[COMMINFO ALTER](#)

## **Attività multicast avanzate**

Utilizzare queste informazioni per informazioni sulle attività di gestione avanzate di IBM MQ Multicast come la configurazione dei file `.ini` e l'interoperabilità con IBM MQ LLM.

Per considerazioni sulla sicurezza in un'installazione multicast, consultare [Sicurezza multicast](#).

## **Collegamento tra domini di pubblicazione / sottoscrizione multicast e non multicast**

Utilizzare queste informazioni per comprendere cosa accade quando un publisher non multicast pubblica in un argomento abilitato a IBM MQ Multicast.

Se un publisher non multicast pubblica un argomento definito come **MCAST** abilitato e **BRIDGE** abilitato, il gestore code trasmette il messaggio tramite multicast direttamente a tutti i sottoscrittori che potrebbero essere in ascolto. Un publisher multicast non può eseguire la pubblicazione su argomenti che non sono abilitati multicast.

Gli argomenti esistenti possono essere abilitati multicast impostando i parametri **MCAST** e **COMMINFO** di un oggetto argomento. Consultare [Initial multicast concepts](#) per ulteriori informazioni su questi parametri.

L'attributo oggetto COMMINFO **BRIDGE** controlla le pubblicazioni dalle applicazioni che non utilizzano multicast. Se **BRIDGE** è impostato su **ENABLED** e il parametro **MCAST** dell'argomento è impostato anche su **ENABLED**, le pubblicazioni delle applicazioni che non utilizzano il multicast vengono collegate tramite bridge alle applicazioni che lo utilizzano. Per ulteriori informazioni sul parametro **BRIDGE**, consultare [DEFINE COMMINFO](#).

## Configurazione dei file .ini per Multicast

Utilizzare queste informazioni per comprendere i campi IBM MQ Multicast nei file .ini .

È possibile eseguire un'ulteriore configurazione di IBM MQ Multicast in un file ini . Il file ini specifico che è necessario utilizzare dipende dal tipo di applicazioni:

- Client: configurare il file `MQ_DATA_PATH/mqclient.ini` .
- Gestore code: configurare il file `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini` .

dove `MQ_DATA_PATH` è l'ubicazione della directory di dati IBM MQ ( `/var/mqm/mqclient.ini` ) e `QMNAME` è il nome del gestore code a cui si applica il file .ini .

Il file .ini contiene campi utilizzati per ottimizzare il comportamento di IBM MQ Multicast:

```
Multicast:
Protocol      = IP | UDP
IPVersion     = IPv4 | IPv6 | ANY | BOTH
LimitTransRate = DISABLED | STATIC | DYNAMIC
TransRateLimit = 100000
SocketTTL     = 1
Batch         = NO
Loop          = 1
Interface     = <IPAddress>
FeedbackMode  = ACK | NACK | WAIT1
HeartbeatTimeout = 20000
HeartbeatInterval = 2000
```

### Protocollo

#### UDP

In questa modalità, i pacchetti vengono inviati utilizzando il protocollo UDP. Tuttavia, gli elementi di rete non possono fornire assistenza nella distribuzione multicast come avviene in modalità IP. Il formato del pacchetto rimane compatibile con PGM. Questo è il valore predefinito.

#### IP

In questa modalità, il trasmettitore invia pacchetti IP non elaborati. Gli elementi di rete con supporto PGM assistono nella distribuzione di pacchetti multicast affidabili. Questa modalità è completamente compatibile con lo standard PGM.

### IPVERSION

#### IPv4

Comunicare solo utilizzando il protocollo IPv4 . Questo è il valore predefinito.

#### IPv6

Comunicare solo utilizzando il protocollo IPv6 .

#### ANY

Comunicare utilizzando IPv4, IPv6 entrambi, a seconda del protocollo disponibile.

#### ENTRAMBI

Supporta la comunicazione utilizzando IPv4 e IPv6.

### Frequenza LimitTrans

#### DISABILITATO

Non esiste alcun controllo della velocità di trasmissione. Questo è il valore predefinito.

#### STATICO

Implementa il controllo della velocità di trasmissione statica. Il trasmettitore non trasmetterà ad una velocità superiore a quella specificata dal parametro `TransRateLimit`.

#### DINAMICO

Il trasmettitore adatta la sua velocità di trasmissione in base al feedback che riceve dai ricevitori. In questo caso, il limite della velocità di trasmissione non può essere superiore al valore specificato dal parametro `TransRateLimit`. Il trasmettitore cerca di raggiungere una velocità di trasmissione ottimale.

### Limite TransRate

Il limite di velocità di trasmissione in Kbps.

### SocketTTL

Il valore di SocketTTL determina se il traffico multicast può passare attraverso un router o il numero di router che può passare attraverso.

### Batch

Controlla se i messaggi vengono batch o inviati immediatamente. Ci sono 2 valori possibili:

- *NO* I messaggi non sono in batch, vengono inviati immediatamente.
- *Si* I messaggi vengono sottoposti a batch.

### Loop

Impostare il valore su 1 per abilitare il loop multicast. Il loop multicast definisce se i dati inviati vengono inviati in loop o meno all'host.

### Interfaccia

L'indirizzo IP dell'interfaccia su cui fluisce il traffico multicast. Per ulteriori informazioni e per la risoluzione dei problemi, consultare: [Test delle applicazioni multicast su una rete non multicast e Impostazione della rete appropriata per il traffico multicast](#)

### FeedbackMode

#### NACK

Feedback da conferme negative. Questo è il valore predefinito.

#### ACK

Feedback da riconoscimenti positivi.

#### WAIT1

Feedback da parte di riconoscimenti positivi in cui il trasmettitore attende solo 1 ACK da uno qualsiasi dei ricevitori.

### HeartbeatTimeout

Il timeout di heartbeat in millisecondi. Il valore 0 indica che gli eventi di timeout heartbeat non vengono generati dal destinatario o dai destinatari dell'argomento. Il valore predefinito è 20000.

### HeartbeatInterval

L'intervallo di heartbeat in millisecondi. Il valore 0 indica che non viene inviato alcun heartbeat. L'intervallo di heartbeat deve essere notevolmente inferiore al valore **HeartbeatTimeout** per evitare eventi di timeout di heartbeat falsi. Il valore predefinito è 2000.

## Interoperabilità multicast con IBM MQ Low Latency Messaging

Utilizzare queste informazioni per comprendere l'interoperabilità tra IBM MQ Multicast e IBM MQ Low Latency Messaging (LLM).

Il trasferimento payload di base è possibile per un'applicazione che utilizza LLM, con un'altra applicazione che utilizza multicast per scambiare messaggi in entrambe le direzioni. Sebbene multicast utilizzi la tecnologia LLM, il prodotto LLM stesso non è incorporato. Pertanto è possibile installare sia LLM che IBM MQ Multicast, e operare e servire i due prodotti separatamente.

Le applicazioni LLM che comunicano con multicast potrebbero dover inviare e ricevere le proprietà del messaggio. Le proprietà del messaggio IBM MQ e i campi MQMD vengono trasmessi come proprietà del messaggio LLM con codici di proprietà del messaggio LLM specifici come mostrato nella tabella seguente:

IBM MQ proprietà	Tipo di proprietà IBM MQ LLM	Tipo di proprietà LLM	Codice proprietà LLM
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003

Tabella 20. Associazioni delle proprietà IBM MQ message to IBM MQ LLM (Continua)

IBM MQ proprietà	Tipo di proprietà IBM MQ LLM	Tipo di proprietà LLM	Codice proprietà LLM
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	Stringa LLM_PROP_KIND_	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	Stringa LLM_PROP_KIND_	-1013
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	Stringa LLM_PROP_KIND_	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	Stringa LLM_PROP_KIND_	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	Stringa LLM_PROP_KIND_	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	Stringa LLM_PROP_KIND_	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

Per ulteriori informazioni su LLM, consultare la documentazione del prodotto LLM: [IBM MQ Low Latency Messaging](#).

## IBM i Amministrazione IBM MQ for IBM i

Introduce i metodi disponibili per amministrare IBM MQ su IBM i.

Le attività di gestione includono la creazione, l'avvio, la modifica, la visualizzazione, l'arresto e la cancellazione di cluster, processi e oggetti IBM MQ (gestori code, code, elenchi nomi, definizioni di processo, canali, canali di connessione client, listener, servizi e oggetti di informazioni di autenticazione).

Consultare i seguenti link per dettagli su come amministrare IBM MQ for IBM i:

- [“Gestione di IBM MQ for IBM i utilizzando i comandi CL” a pagina 344](#)
- [“Modi alternativi di amministrazione di IBM MQ for IBM i” a pagina 357](#)
- [“Gestione del lavoro per IBM i” a pagina 363](#)

### Concetti correlati

[“Disponibilità, backup, ripristino e riavvio su IBM i” a pagina 370](#)

Utilizzare queste informazioni per comprendere in che modo IBM MQ for IBM i utilizza il supporto journal IBM i per facilitare la propria strategia di backup e ripristino.

[Informazioni sui nomi delle librerie dei gestori code IBM MQ for IBM i](#)

### Attività correlate

[Modifica delle informazioni di configurazione su IBM i](#)

[Impostazione della sicurezza su IBM i](#)

### Riferimenti correlati

[“In corso di quiesce IBM MQ for IBM i” a pagina 414](#)

Questa sezione spiega come disattivare (terminare correttamente) IBM MQ for IBM i.

[“Il gestore code di messaggi non instradabili su IBM i.” a pagina 149](#)

Cos'è il gestore code di messaggi non recapitabili IBM i e come viene richiamato?

## Gestione di IBM MQ for IBM i utilizzando i comandi CL

Utilizzare queste informazioni per comprendere i comandi IBM MQ IBM i .

È possibile accedere alla maggior parte dei gruppi di comandi IBM MQ , inclusi quelli associati ai gestori code, alle code, agli argomenti, ai canali, agli elenchi nomi, alle definizioni dei processi e agli oggetti delle informazioni di autenticazione utilizzando il comando **WRK\*** pertinente.

Il comando principale nella serie è **WRKMQM**. Questo comando consente, ad esempio, di visualizzare un elenco di tutti i gestori code sul sistema, insieme alle informazioni sullo stato. In alternativa, è possibile elaborare tutti i comandi specifici del gestore code utilizzando diverse opzioni per ciascuna voce.

Dal comando **WRKMQM** è possibile selezionare aree specifiche di ciascun gestore code, ad esempio, gestione di canali, argomenti o code e da lì selezionare singoli oggetti.

### Registrazione delle definizioni dell'applicazione IBM MQ

Quando si creano o personalizzano applicazioni IBM MQ , è utile conservare un record di tutte le definizioni IBM MQ create. Questo record può essere utilizzato per:

- Scopi di recupero
- Manutenzione
- Rollout delle applicazioni IBM MQ

È possibile registrare le definizioni di applicazione IBM MQ in 1 di 2 modi:

1. Creazione di programmi CL per creare le definizioni IBM MQ per il server.
2. Creazione di file di testo MQSC come membri SRC per generare le definizioni IBM MQ utilizzando il linguaggio di comandi IBM MQ multiplatforma.

Per ulteriori dettagli sulla definizione degli oggetti coda, consultare [“Gestione mediante comandi MQSC”](#) a pagina 11 e [“Utilizzo di IBM MQ Programmable Command Format”](#) a pagina 25.

#### Riferimenti correlati

[IBM MQ for IBM i Riferimento comandi CL](#)

## Prima di iniziare ad utilizzare IBM MQ for IBM i utilizzando i comandi CL

Utilizzare queste informazioni per avviare il sottosistema IBM MQ e creare un gestore code locale.

### Prima di iniziare

Assicurarsi che il sottosistema IBM MQ sia in esecuzione (utilizzando il comando STRSBS QMQM/QMQM ) e che la coda lavori associata a tale sottosistema non sia congelata. Per impostazione predefinita, il sottosistema IBM MQ e la coda lavori sono entrambi denominati QMQM nella libreria QMQM.

### Informazioni su questa attività

Utilizzo della riga comandi IBM i per avviare un gestore code

### Procedura

1. Creare un gestore code locale emettendo il comando CRTMQM da una riga comandi IBM i .  
Quando si crea un gestore code, è possibile impostare tale gestore code come gestore code predefinito. Il gestore code predefinito (di cui può essere presente solo uno) è il gestore code a cui si applica un comando CL, se il parametro del nome del gestore code (MQMNAME) viene omesso.

2. Avviare il gestore code locale emettendo il comando STRMQM da una riga comandi IBM i .

Se l'avvio del gestore code impiega più di alcuni secondi, IBM MQ visualizzerà i messaggi di stato in modo intermittente che descrivono in dettaglio l'avanzamento dell'avvio. Per ulteriori informazioni su questi messaggi, consultare [Messaggi e codici di errore](#).

## Operazioni successive

È possibile arrestare un gestore code immettendo il comando ENDMQM dalla riga comandi IBM i e controllare un gestore code immettendo altri comandi IBM MQ da una riga comandi IBM i .

I gestori code remoti non possono essere avviati in remoto, ma devono essere creati e avviati nei loro sistemi da operatori locali. Un'eccezione è rappresentata dalle funzioni operative remote (all'esterno di IBM MQ for IBM i) per abilitare tali operazioni.

L'amministratore della coda locale non può arrestare un gestore code remoto.

**Nota:** Come parte della disattivazione di un sistema IBM MQ , è necessario disattivare i gestori code attivi. Ciò è descritto in ["In corso di quiesce IBM MQ for IBM i"](#) a pagina 414.

## IBM i Creazione di oggetti IBM MQ for IBM i

Utilizzare queste informazioni per comprendere i metodi per la creazione di oggetti IBM MQ per IBM i.

### Prima di iniziare

Le seguenti attività suggeriscono diversi modi in cui è possibile utilizzare IBM MQ for IBM i dalla riga comandi.

### Informazioni su questa attività

Ci sono due metodi online per creare oggetti IBM MQ , che sono:

### Procedura

1. Utilizzando un comando Create, ad esempio: il comando **Create MQM Queue CRTMQMQ**
2. Utilizzo di un comando di gestione oggetti MQM, seguito da F6, ad esempio: il comando **Work with MQM Queues : WRKMQMQ**

### Operazioni successive

Per un elenco di tutti i comandi consultare [IBM MQ for IBM i Comandi CL](#).

**Nota:** Tutti i comandi MQM possono essere inoltrati dal menu Comandi gestore code messaggi. Per visualizzare questo menu, immettere GO CDMQM sulla riga comandi e premere il tasto Enter .

Il sistema visualizza automaticamente il pannello di richiesta quando si seleziona un comando da questo menu. Per visualizzare il pannello di richiesta per un comando immesso direttamente sulla riga comandi, premere F4 prima di premere il tasto Enter .

### **Creazione di una coda locale utilizzando il comando CRTMQMQ**

### Procedura

1. Immettere CHGMQM sulla riga comandi e premere il tasto F4 .
2. Nel pannello **Crea coda MQM**, immettere il nome della coda che si desidera creare nel campo Queue name . Per specificare un nome con maiuscole e minuscole, racchiudere il nome tra apici.
3. Immettere \*LCL nel campo Queue type .
4. Specificare un nome gestore code, a meno che non si utilizzi il gestore code predefinito e premere il tasto Enter . È possibile sovrascrivere qualsiasi valore con un nuovo valore. Scorrere in avanti per visualizzare ulteriori campi. Le opzioni utilizzate per i cluster sono alla fine dell'elenco di opzioni.

5. Una volta modificati i valori, premere il pulsante Enter per creare la coda.

## **Creazione di una coda locale utilizzando il comando WRKMQMQ**

### **Procedura**

1. Immettere WRKMQMQ sulla riga comandi.
2. Immettere il nome di un Gestore code.
3. Se si desidera visualizzare il pannello di richiesta, premere F4. Il pannello di richiesta è utile per ridurre il numero di code visualizzate, specificando un nome coda generico o un tipo di coda.
4. Premere Enter e viene visualizzato il pannello **Gestisci code MQM** . È possibile sovrascrivere qualsiasi valore con un nuovo valore. Scorrere in avanti per visualizzare ulteriori campi. Le opzioni utilizzate per i cluster sono alla fine dell'elenco di opzioni.
5. Premere F6 per creare una nuova coda; viene visualizzato il pannello **CRTMQMQ** . Consultare [“Creazione di una coda locale utilizzando il comando CRTMQMQ”](#) a pagina 345 per istruzioni su come creare la coda. Una volta creata la coda, viene visualizzato nuovamente il pannello **Gestisci code MQM** . La nuova coda viene aggiunta all'elenco quando si preme F5=Refresh.

### **Modifica degli attributi del gestore code**

#### **Informazioni su questa attività**

Per modificare gli attributi del gestore code specificato sul comando **CHGMQM** , specificando attributi e valori che si desidera modificare. Ad esempio, utilizzare le seguenti opzioni per modificare gli attributi di `jupiter.queue.manager`:

### **Procedura**

Immettere **CHGMQM** sulla riga comandi e premere il tasto F4 .

### **Risultati**

Il comando modifica la coda di messaggi non instradabili utilizzata e abilita gli eventi di inibizione.

## **Gestione delle code locali su IBM i**

Questa sezione contiene esempi di alcuni comandi che è possibile utilizzare per gestire le code locali. Tutti i comandi visualizzati sono disponibili anche utilizzando le opzioni dal **pannello dei comandi WRKMQMQ**.

### **Definizione di una coda locale**

Per un'applicazione, il gestore code locale è il gestore code a cui è connessa l'applicazione. Le code gestite da un gestore code locale vengono dette locali per tale gestore code.

Utilizzare il comando **CRTMQMQ QTYPE \*LCL** per creare una definizione di una coda locale e anche per creare la struttura dati denominata coda. È anche possibile modificare le caratteristiche della coda da quelle della coda locale predefinita.

In questo esempio, la coda definita, `orange.local.queue`, è specificata per avere le seguenti caratteristiche:

- È abilitato per le ricezioni, disabilitato per le inserimenti e opera su base FIFO (first - in - first - out).
- È una coda *ordinaria* , vale a dire, non è una coda di iniziazione o una coda di trasmissione e non genera messaggi trigger.
- La lunghezza massima della coda è 1000 messaggi; la lunghezza massima del messaggio è 2000 byte.

Il seguente comando esegue questa operazione sul gestore code predefinito:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL)
TEXT('Queue for messages from other systems')
PUTENBL(*NO)
GETENBL(*YES)
TRGENBL(*NO)
MSGDLYSEQ(*FIFO)
MAXDEPTH(1000)
MAXMSGLEN(2000)
USAGE(*NORMAL)
```

#### Nota:

1. USAGE \*NORMAL indica che questa coda non è una coda di trasmissione.
2. Se si dispone già di una coda locale con il nome `orange.local.queue` sullo stesso gestore code, questo comando non riesce. Utilizzare l'attributo REPLACE \*YES se si desidera sovrascrivere la definizione esistente di una coda, ma consultare anche [“Modifica degli attributi della coda locale”](#) a pagina 348.

## Definizione di una coda di messaggi non recapitabili

Ogni gestore code deve avere una coda locale da utilizzare come coda di messaggi non recapitabili in modo che i messaggi che non possono essere consegnati alla destinazione corretta possano essere memorizzati per un successivo recupero. È necessario informare esplicitamente il gestore code della coda di messaggi non recapitabili. È possibile eseguire questa operazione specificando una coda di messaggi non recapitabili nel comando **CRTMQMQ** oppure è possibile utilizzare il comando **CHGMQM** per specificarne una in un secondo momento. È inoltre necessario definire la coda di messaggi non recapitabili prima che possa essere utilizzata.

Una coda di messaggi non recapitabili di esempio denominata `SYSTEM.DEAD.LETTER.QUEUE` viene fornita con il prodotto. Questa coda viene creata automaticamente quando si crea il gestore code. Se necessario, è possibile modificare questa definizione. Non c'è bisogno di rinominarlo, anche se si può, se ti piace.

Una coda di messaggi non recapitabili non ha requisiti speciali tranne che:

- Deve essere una coda locale.
- Il suo attributo MAXMSGL (lunghezza massima messaggio) deve consentire alla coda di contenere i messaggi più grandi che il gestore code deve gestire **più** la dimensione dell'intestazione dei messaggi non instradabili (MQDLH).

IBM MQ fornisce un gestore code di messaggi non recapitabili che consente di specificare il modo in cui i messaggi trovati in una coda di messaggi non recapitabili devono essere elaborati o rimossi. Per ulteriori informazioni, fare riferimento a [“Il gestore code di messaggi non instradabili su IBM i.”](#) a pagina 149.

## Visualizzazione degli attributi dell'oggetto predefiniti

Quando si definisce un oggetto IBM MQ, vengono utilizzati tutti gli attributi non specificati dall'oggetto predefinito. Ad esempio, quando si definisce una coda locale, la coda eredita tutti gli attributi omissi nella definizione dalla coda locale predefinita, denominata `SYSTEM.DEFAULT.LOCAL.QUEUE`. Per vedere esattamente quali sono questi attributi, utilizzare il seguente comando:

```
DSPMQMQ QNAME(SYSTEM.DEFAULT.LOCAL.QUEUE) MQMNAME(MYQUEUEMANAGER)
```

## Copia di una definizione di coda locale

È possibile copiare una definizione di coda utilizzando il comando `CPYMQMQ`. Ad esempio:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Questo comando crea una coda con gli stessi attributi della coda originale `orange.local.queue`, piuttosto che quelli della coda locale predefinita del sistema.

È anche possibile utilizzare il comando **CPYMQMQ** per copiare una definizione di coda, ma sostituendo una o più modifiche agli attributi dell'originale. Ad esempio:

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(1024)
```

Questo comando copia gli attributi della coda `orange.local.queue` nella coda `third.queue`, ma specifica che la lunghezza massima del messaggio sulla nuova coda deve essere di 1024 byte anziché 2000.

**Nota:** Quando si utilizza il comando **CPYMQMQ**, si copiano solo gli attributi della coda, non i messaggi sulla coda.

## Modifica degli attributi della coda locale

È possibile modificare gli attributi della coda in due modi, utilizzando il comando **CHGMQMQ** o il comando **CPYMQMQ** con l'attributo `REPLACE *YES`. In [“Definizione di una coda locale”](#) a pagina 346, è stata definita la coda `orange.local.queue`. Se, ad esempio, è necessario aumentare la lunghezza massima del messaggio su questa coda a 10.000 byte.

- Utilizzando il comando **CHGMQMQ** :

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Questo comando modifica un singolo attributo, quello della lunghezza massima del messaggio; tutti gli altri attributi rimangono uguali.

- Utilizzando il comando **CRTMQMQ** con l'opzione `REPLACE *YES`, ad esempio:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(10000) REPLACE(*YES)
```

Questo comando modifica non solo la lunghezza massima del messaggio, ma anche tutti gli altri attributi, ai quali vengono assegnati valori predefiniti. La coda è ora abilitata all'inserimento, mentre in precedenza era inibita. L'inserimento abilitato è quello predefinito, come specificato dalla coda `SYSTEM.DEFAULT.LOCAL.QUEUE`, a meno che non sia stato modificato.

Se si *riduce* la lunghezza massima dei messaggi su una coda esistente, i messaggi esistenti non vengono influenzati. Qualsiasi nuovo messaggio, tuttavia, deve soddisfare i nuovi criteri.

## Cancellazione di una coda locale

Per eliminare tutti i messaggi da una coda locale denominata `magenta.queue`, utilizzare il seguente comando:

```
CLRMQMQ QNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Non è possibile cancellare una coda se:

- Ci sono messaggi di cui non è stato eseguito il commit che sono stati inseriti nella coda nel punto di sincronizzazione.
- Un'applicazione ha attualmente la coda aperta.

## Eliminazione di una coda locale

Utilizzare il comando **DLTMQMQ** per eliminare una coda locale.

Una coda non può essere eliminata se contiene messaggi di cui non è stato eseguito il commit o se è in uso.

## Abilitazione di code grandi

IBM MQ supporta code più grandi di 2 GB. Consultare la documentazione del sistema operativo per informazioni su come abilitare IBM i per supportare file di grandi dimensioni.

Le informazioni sul prodotto IBM i sono disponibili in [IBM Documentation](#).

Alcuni programmi di utilità potrebbero non essere in grado di gestire i file superiori a 2 GB. Prima di abilitare il supporto file di grandi dimensioni, consultare la documentazione del sistema operativo per informazioni sulle restrizioni su tale supporto.

## IBM i Gestione delle code alias su IBM i

Questa sezione contiene esempi di alcuni comandi che è possibile utilizzare per gestire le code alias. Tutti i comandi visualizzati sono disponibili anche utilizzando le opzioni dal **pannello dei comandi WRKMQMQ**.

Una coda alias (a volte nota come alias della coda) fornisce un metodo di reindirizzamento delle chiamate MQI. Una coda alias non è una coda reale ma una definizione che si risolve in una coda reale. La definizione della coda alias contiene un nome coda di destinazione, specificato dall'attributo TGTQNAME .

Quando un'applicazione specifica una coda alias in una chiamata MQI, il gestore code risolve il nome della coda reale in fase di runtime.

Ad esempio, un'applicazione è stata sviluppata per inserire i messaggi su una coda denominata `my.alias.queue`. Specifica il nome di questa coda quando effettua una richiesta **MQOPEN** e, indirettamente, se inserisce un messaggio su questa coda. L'applicazione non è consapevole che la coda è una coda alias. Per ogni chiamata MQI che utilizza questo alias, il gestore code risolve il nome della coda reale, che potrebbe essere una coda locale o una coda remota definita su questo gestore code.

Modificando il valore dell'attributo TGTQNAME , è possibile reindirizzare chiamate MQI a un'altra coda, possibilmente su un altro gestore code. È utile per la manutenzione, la migrazione e il bilanciamento del carico.

## Definizione di una coda alias

Il seguente comando crea una coda alias:

```
CRTMQMQ QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')  
MQMNAME(MYQUEUEMANAGER)
```

Questo comando reindirizza le chiamate MQI che specificano `my.alias.queue` alla coda `yellow.queue`. Il comando non crea la coda di destinazione; le chiamate MQI hanno esito negativo se la coda `yellow.queue` non esiste al runtime.

Se si modifica la definizione dell'alias, è possibile reindirizzare le chiamate MQI a un'altra coda. Ad esempio:

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Questo comando reindirizza chiamate MQI a un'altra coda, `magenta.queue`.

È inoltre possibile utilizzare le code alias per fare in modo che una singola coda (la coda di destinazione) sembri avere attributi differenti per applicazioni differenti. Questa operazione viene eseguita definendo due alias, uno per ogni applicazione. Si supponga che vi siano due applicazioni:

- L'applicazione ALPHA può inserire messaggi su `yellow.queue`, ma non è consentita la ricezione di messaggi da esso.
- L'applicazione BETA può ricevere messaggi da `yellow.queue`, ma non è consentito inserirlo.

È possibile eseguire questa operazione utilizzando i comandi riportati di seguito:

```
/* This alias is put enabled and get disabled for application ALPHA */
CRTMQMQ QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQMQ QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEMANAGER)
```

ALPHA utilizza il nome della coda `alphas.alias.queue` nelle relative chiamate MQI; BETA utilizza il nome della coda `betas.alias.queue`. Entrambi accedono alla stessa coda, ma in modi diversi.

È possibile utilizzare l'attributo `REPLACE *YES` quando si definiscono le code alias, nello stesso modo in cui si utilizzano questi attributi con le code locali.

## Utilizzo di altri comandi con code alias

È possibile utilizzare i comandi appropriati per visualizzare o modificare gli attributi della coda alias. Ad esempio:

```
* Display the alias queue's attributes */
DSPMQMQ QNAME('alphas.alias.queue') MQMNAME(MYQUEUEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */
CHQMCMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEMANAGER)
```



## Utilizzo delle code modello su IBM i

Questa sezione contiene esempi di alcuni dei comandi che è possibile utilizzare per gestire le code modello. Tutti i comandi visualizzati sono disponibili anche utilizzando le opzioni dal **pannello dei comandi WRKMQMQ**.

Un gestore code crea una coda dinamica se riceve una chiamata MQI da un'applicazione che specifica un nome coda che è stato definito come una coda modello. Il nome della nuova coda dinamica viene generato dal gestore code quando viene creata la coda. Una coda modello è un modello che specifica gli attributi di qualsiasi coda dinamica da essa creata.

Le code modello forniscono un metodo conveniente per le applicazioni per creare le code come sono richieste.

## Definizione di una coda modello

Si definisce una coda modello con un insieme di attributi nello stesso modo in cui si definisce una coda locale. Le code modello e le code locali hanno la stessa serie di attributi, tranne per il fatto che sulle code modello è possibile specificare se le code dinamiche create sono temporanee o permanenti. (Le code permanenti vengono conservate durante i riavvii del gestore code, mentre quelle temporanee non lo sono). Ad esempio:

```
CRTMQMQ QNAME('green.model.queue') QTYPE(*MDL) DFNTYPE(*PERMDYN)
```

Questo comando crea una definizione di coda modello. Dall'attributo `DFNTYPE`, le code effettive create da questo template sono code dinamiche permanenti. Gli attributi non specificati vengono copiati automaticamente dalla coda predefinita `SYSYSTEM.DEFAULT.MODEL.QUEUE`.

È possibile utilizzare l'attributo `REPLACE *YES` quando si definiscono le code modello, nello stesso modo in cui si utilizzano le code locali.

## Utilizzo di altri comandi con code modello

È possibile utilizzare i comandi appropriati per visualizzare o modificare gli attributi di una coda modello. Ad esempio:

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQM MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

## IBM i Utilizzo del trigger su IBM i

Utilizzare queste informazioni per informazioni sul trigger e sulle definizioni di processo.

IBM MQ fornisce una funzione per avviare automaticamente un'applicazione quando vengono soddisfatte determinate condizioni su una coda. Un esempio di condizioni è quando il numero di messaggi su una coda raggiunge un numero specificato. Questa funzionalità è denominata *attivazione* ed è descritta in modo dettagliato in [Trigger canali](#).

### Cos'è l'attivazione?

Il gestore code definisce alcune condizioni come eventi trigger. Se il trigger è abilitato per una coda e si verifica un evento trigger, il gestore code invia un messaggio trigger a una coda denominata coda di iniziazione. La presenza del messaggio trigger sulla coda di iniziazione indica che si è verificato un evento trigger.

I messaggi trigger generati dal gestore code non sono persistenti. Ciò ha l'effetto di ridurre la registrazione (migliorando quindi le prestazioni) e di ridurre i duplicati durante il riavvio, in modo da migliorare il tempo di riavvio.

### Cos'è il controllo trigger?

Il programma che elabora la coda di iniziazione viene chiamato un'applicazione di controllo dei trigger e la sua funzione è di leggere il messaggio di trigger e intraprendere l'azione appropriata, in base alle informazioni contenute nel messaggio di trigger. Di solito, questa azione è quella di avviare un'altra applicazione per elaborare la coda che ha causato la generazione del messaggio trigger. Dal punto di vista del gestore code, non c'è nulla di speciale nell'applicazione di controllo dei trigger - è un'altra applicazione che legge i messaggi da una coda (la coda di avvio).

### Modifica degli attributi di inoltro del lavoro del controllo trigger

Il controllo trigger fornito come comando **STRMQMTRM** inoltra un lavoro per ogni messaggio trigger utilizzando la descrizione lavoro predefinita di sistema, QDFTJOB. Ciò ha delle limitazioni in quanto i lavori inoltrati vengono sempre chiamati QDFTJOB e hanno gli attributi della descrizione lavoro predefinita incluso l'elenco librerie, \*SYSVAL. IBM MQ fornisce un metodo per sovrascrivere questi attributi. Ad esempio, è possibile personalizzare i lavori inoltrati in modo che abbiano nomi di lavoro più significativi come segue:

1. Nella descrizione del lavoro specificare la descrizione desiderata, ad esempio i valori di registrazione.
2. Specificare i dati di ambiente della definizione di processo utilizzata nel processo di attivazione:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

Il controllo trigger esegue un SBMJOB utilizzando la descrizione specificata.

È possibile sovrascrivere altri attributi di SBMJOB specificando la parola chiave e il valore appropriati nei dati di ambiente della definizione del processo. L'unica eccezione è la parola chiave CMD perché questo

attributo è riempito dal controllo trigger. Di seguito è riportato un esempio del comando per specificare i dati di ambiente della definizione del processo in cui devono essere modificati sia il nome del lavoro che la descrizione:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)
JOB(TRIGGER)')
```

## Definizione di una coda dell'applicazione per il trigger

Una coda di applicazione è una coda locale utilizzata dalle applicazioni per la messaggistica, tramite MQI. L'attivazione richiede la definizione di un certo numero di attributi della coda sulla coda dell'applicazione. L'attivazione è abilitata dall'attributo TRGENBL .

In questo esempio, un evento trigger deve essere generato quando ci sono 100 messaggi di priorità 5 o superiore sulla coda locale `motor.insurance.queue`, come segue:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)
PRCNAME('motor.insurance.quote.process') MAXMSGLEN(2000)
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')
TRGENBL(*YES) TRGTYPE(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

dove i parametri sono:

### **MQMNAME (MYQUEUEMANAGER)**

Il nome del gestore code.

### **QNAME ('motor.insurance.queue')**

Il nome della coda dell'applicazione definita.

### **PRCNAME ('motor.insurance.quote.process')**

Il nome dell'applicazione che deve essere avviata da un programma di controllo trigger.

### **MAXMSGLEN (2000)**

La lunghezza massima dei messaggi sulla coda.

### **DFTMSGPST (\*YES)**

I messaggi su questa coda sono persistenti per impostazione predefinita.

### **INITQNAME ('motor.ins.init.queue')**

Il nome della coda di avvio in cui il gestore code deve inserire il messaggio trigger.

### **TRGENBL (\*YES)**

Il valore dell'attributo trigger.

### **TRGTYPE (\*DEPTH)**

Un evento trigger viene generato quando il numero di messaggi con la priorità richiesta ( **TRGMSGPTY** ) raggiunge il numero specificato in **TRGDEPTH**.

### **TRGDEPTH (100)**

Il numero di messaggi richiesti per generare un evento trigger.

### **TRGMSGPTY (5)**

La priorità dei messaggi che devono essere conteggiati dal gestore code per decidere se generare un evento trigger. Vengono contati solo i messaggi con priorità 5 o superiore.

## Definizione di una coda di iniziazione

Quando si verifica un evento trigger, il gestore code inserisce un messaggio trigger sulla coda di avvio specificata nella definizione della coda dell'applicazione. Le code di iniziazione non dispongono di impostazioni speciali, ma è possibile utilizzare la seguente definizione della coda locale `motor.ins.init.queue` come guida:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.ins.init.queue') QTYPE(*LCL)
GETENBL(*YES) SHARE(*NO) TRGTYPE(*NONE)
```

## Creazione di una definizione di processo

Utilizzare il comando **CRTMQMPCRC** per creare una definizione di processo. Una definizione di processo associa una coda dell'applicazione all'applicazione che deve elaborare i messaggi dalla coda. Ciò viene eseguito tramite l'attributo **PRCNAME** nella coda dell'applicazione `motor.insurance.queue`. Il seguente comando crea il processo richiesto, `motor.insurance.quote.process`, identificato in questo esempio:

```
CRTMQMPCRC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')  
TEXT('Insurance request message processing')  
APPTYPE(*OS400) APPID(MQTEST/TESTPROG)  
USRDATA('open, close, 235')
```

dove i parametri sono:

### **MQMNAME(MYQUEUEMANAGER)**

Il nome del gestore code.

### **PRCNAME('motor.insurance.quote.process')**

Il nome della definizione del processo.

### **TEXT('Insurance request message processing')**

Una descrizione del programma applicativo a cui si riferisce questa definizione. Questo testo viene visualizzato quando si utilizza il comando **DSPMQMPCRC**. Ciò consente di identificare le operazioni del processo. Se si utilizzano spazi nella stringa, è necessario racchiudere la stringa tra virgolette singole.

### **APPTYPE(\*OS400)**

Il tipo di applicazione da avviare.

### **APPID(MQTEST/TESTPROG)**

Il nome del file eseguibile dell'applicazione, specificato come nome file completo.

### **USRDATA('open, close, 235')**

Dati definiti dall'utente, che possono essere utilizzati dall'applicazione.

## Visualizzazione della definizione del processo

Utilizzare il comando **DSPMQMPCRC** per esaminare i risultati della definizione. Ad esempio:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPCRC('motor.insurance.quote.process')
```

È inoltre possibile utilizzare il comando **CHGMQMPCRC** per modificare una definizione di processo esistente e il comando **DLTMQMPCRC** per eliminare una definizione di processo.

## Comunicazione tra due sistemi IBM MQ su IBM i

Questo esempio di codifica illustra come impostare due sistemi IBM MQ for IBM i, utilizzando comandi CL, in modo che possano comunicare tra loro.

I sistemi sono denominati SYSTEMA e SYSTEMBe il protocollo di comunicazione utilizzato è TCP/IP.

Effettuare la seguente procedura:

1. Creare un gestore code su SYSTEMA, richiamandolo QMGRA1.

```
CRTMQM    MQMNAME(QMGRA1) TEXT('System A - Queue +  
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Avviare questo gestore code.

```
STRMQM MQMNAME(QMGRA1)
```

3. Definire gli IBM MQ oggetti su SYSTEMA necessari per l'invio di messaggi a un gestore code su SYSTEMB.

```
/* Transmission queue */
CRTMQMQ QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +
MQMNAME(QMGRA1) TEXT('Transmission Queue +
to QMGRB1') MAXDEPTH(5000) USAGE(*TMQ)

/* Remote queue that points to a queue called TARGETB */
/* TARGETB belongs to queue manager QMGRB1 on SYSTEMB */
CRTMQMQ QNAME(TARGETB.ON.QMGRB1) QTYPE(*RMT) +
MQMNAME(QMGRA1) TEXT('Remote Q pointing +
at Q TARGETB on QMGRB1 on Remote System +
SYSTEMB') RMTQNAME(TARGETB) +
RMTMQNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)

/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +
MQMNAME(QMGRA1) TRPTYPE(*TCP) +
TEXT('Sender Channel From QMGRA1 on +
SYSTEMA to QMGRB1 on SYSTEMB') +
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. Creare un gestore code su SYSTEMB, richiamandolo QMGRB1.

```
CRTMQM MQMNAME(QMGRB1) TEXT('System B - Queue +
Manager 1') UDLMMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

5. Avviare il gestore code su SYSTEMB.

```
STRMQM MQMNAME(QMGRB1)
```

6. Definire gli oggetti IBM MQ necessari per ricevere i messaggi dal gestore code su SYSTEMA.

```
/* Local queue to receive messages on */
CRTMQMQ QNAME(TARGETB) QTYPE(*LCL) MQMNAME(QMGRB1) +
TEXT('Sample Local Queue for QMGRB1')

/* Receiver channel of the same name as the sender channel on SYSTEMA */
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*RCVR) +
MQMNAME(QMGRB1) TRPTYPE(*TCP) +
TEXT('Receiver Channel from QMGRA1 to +
QMGRB1')
```

7. Infine, avviare un listener TCP/IP su SYSTEMB in modo che il canale possa essere avviato. Questo esempio utilizza la porta predefinita 1414.

```
STRMQMLSR MQMNAME(QMGRB1)
```

È ora possibile inviare i messaggi di verifica tra SYSTEMA e SYSTEMB. Utilizzando uno degli esempi forniti, inserire una serie di messaggi nella coda remota su SYSTEMA.

Avviare il canale su SYSTEMA, utilizzando il comando **STRMQMCHL** oppure utilizzando il comando **WRKMQMCHL** e immettendo una richiesta di avvio (Opzione 14) rispetto al canale mittente.

Il canale deve passare allo stato RUNNING e i messaggi vengono inviati alla coda TARGETB su SYSTEMB.

Controllare i messaggi immettendo il comando:

```
WRKMQMMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

Questo esempio contiene il programma CL AMQSAMP4 sample IBM i.

```

/*****/
/*
/* Program name: AMQSAMP4
/*
/* Description: Sample CL program defining MQM queues
/* to use with the sample programs
/* Can be run, with changes as needed, after
/* starting the MQM
/*
/* <N_OCO_COPYRIGHT> */
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1993, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT> */
/*
/*****/
/*
/* Function:
/*
/* AMQSAMP4 is a sample CL program to create or reset the
/* MQI resources to use with the sample programs.
/*
/* This program, or a similar one, can be run when the MQM
/* is started - it creates the objects if missing, or resets
/* their attributes to the prescribed values.
/*
/*
/*
/* Exceptions signaled: none
/* Exceptions monitored: none
/*
/* AMQSAMP4 takes a single parameter, the Queue Manager name
/*
/*****/
QSYS/PGM PARM(&QMGRNAME)

/*****/
/* Queue Manager Name Parameter
/*****/
QSYS/DCL VAR(&QMGRNAME) TYPE(*CHAR)

/*****/
/* EXAMPLES OF DIFFERENT QUEUE TYPES
/*
/* Create local, alias and remote queues
/*
/* Uses system defaults for most attributes
/*
/*****/
/* Create a local queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.LOCAL') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Sample local queue') /* description */+
SHARE(*YES) /* Shareable */+
DFTMSGPST(*YES) /* Persistent messages OK */

/* Create an alias queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ALIAS') +
MQMNAME(&QMGRNAME) +
QTYPE(*ALS) REPLACE(*YES) +
+
TEXT('Sample alias queue') +
DFTMSGPST(*YES) /* Persistent messages OK */+
TGTONAME('SYSTEM.SAMPLE.LOCAL')

/* Create a remote queue - in this case, an indirect reference */

```

```

/* is made to the sample local queue on OTHER queue manager */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REMOTE') +
MQMNAME(&QMGRNAME) +
QTYPE(*RMT) REPLACE(*YES) +
+
TEXT('Sample remote queue')/* description */+
DFTMSGPST(*YES) /* Persistent messages OK */+
RMTQNAME('SYSTEM.SAMPLE.LOCAL') +
RMTMQMNAME(OTHER) /* Queue is on OTHER */

/* Create a transmission queue for messages to queues at OTHER */
/* By default, use remote node name */
CRTMQMQ QNAME('OTHER') /* transmission queue name */+
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Transmission queue to OTHER') +
USAGE(*TMQ) /* transmission queue */

/*****
/* SPECIFIC QUEUES AND PROCESS USED BY SAMPLE PROGRAMS */
/* */
/* Create local queues used by sample programs */
/* Create MQI process associated with sample initiation queue */
/* */
/*****
/* General reply queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REPLY') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('General reply queue') +
DFTMSGPST(*NO) /* Not Persistent */

/* Queue used by AMQSINQ4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.INQ') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSINQ4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO) /* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSSET4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.SET') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSSET4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSECH4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ECHO') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSECH4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Initiation Queue used by AMQSTRG4, sample trigger process */
CRTMQMQ QNAME('SYSTEM.SAMPLE.TRIGGER') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Trigger queue for sample programs')

/* MQI Processes associated with triggered sample programs */

```

```

/*
/****** Note - there are versions of the triggered samples *****/
/****** in different languages - set APPID for these *****/
/****** process to the variation you want to trigger *****/
/*
CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
+
TEXT('Trigger process for AMQSINQ4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSINQ4') /* C */ +
/* APPID('QMOM/AMQOINQ4') /* COBOL */ +
/* APPID('QMOM/AMQ3INQ4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
+
TEXT('Trigger process for AMQSSET4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSSET4') /* C */ +
/* APPID('QMOM/AMQOSET4') /* COBOL */ +
/* APPID('QMOM/AMQ3SET4') /* RPG - ILE */

CRTMQMPRC PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
MQMNAME(&QMGRNAME) +
REPLACE(*YES) +
+
TEXT('Trigger process for AMQSECH4') +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSECH4') /* C */ +
/* APPID('QMOM/AMQOECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****
/*
/* Normal return.
/*
/*****
SNDPGMMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM

/*****
/*
/* END OF AMQSAMP4
/*
/*****

```

## IBM i Modi alternativi di amministrazione di IBM MQ for IBM i

L'utilizzo dei comandi CL è il metodo preferito per la gestione di IBM MQ for IBM i. Tuttavia, è possibile utilizzare vari altri metodi di gestione, inclusi i comandi MQSC, PCF e la gestione remota.

Di solito, si utilizzano i comandi CL IBM i per gestire IBM MQ for IBM i. Per una panoramica di questi comandi, consultare [“Gestione di IBM MQ for IBM i utilizzando i comandi CL”](#) a pagina 344.

È possibile utilizzare gli eventi di strumentazione IBM MQ per monitorare il funzionamento dei gestori code. Consultare [Eventi di strumentazione](#) per informazioni sugli eventi di strumentazione IBM MQ e su come utilizzarli.

È possibile utilizzare uno qualsiasi dei metodi di gestione descritti nei seguenti argomenti secondari come alternativa all'utilizzo dei comandi CL IBM i :

### IBM i Amministrazione locale e remota su IBM i

Gli oggetti IBM MQ for IBM i vengono gestiti localmente o in remoto.

Per *amministrazione locale* si intende l'esecuzione di attività di amministrazione su qualsiasi gestore code definito sul sistema locale. In IBM MQ, è possibile considerarlo come amministrazione locale poiché non

sono coinvolti canali IBM MQ , ossia la comunicazione è gestita dal sistema operativo. Per eseguire questo tipo di attività, è necessario accedere al sistema remoto e immettere i comandi da lì oppure creare un processo che possa emettere i comandi per l'utente.

IBM MQ supporta la gestione da un singolo punto tramite la *amministrazione remota*. La gestione remota consiste nell'inviare messaggi di controllo PCF (Programmable Command Format) a SYSTEM.ADMIN.COMMAND.QUEUE sul gestore code di destinazione.

Esistono diversi modi per generare messaggi PCF. Sono:

1. Scrittura di un programma utilizzando messaggi PCF. Consultare [“Gestione mediante comandi PCF su IBM i”](#) a pagina 359.
2. Scrittura di un programma utilizzando MQAI, che invia messaggi PCF. Consultare [“Utilizzo di MQAI per semplificare l'utilizzo dei PCF”](#) a pagina 36.
3. Utilizzando IBM MQ Explorer, disponibile con IBM MQ for Windows, che consente di utilizzare una GUI (graphical user interface) e genera i messaggi PCF corretti. Consultare [“Utilizzo di IBM MQ Explorer con IBM MQ for IBM i”](#) a pagina 360.
4. Utilizzare **STRMQMMQSC** per inviare comandi indirettamente a un gestore code remoto. Consultare [“Gestione utilizzando i comandi MQSC su IBM i”](#) a pagina 358.

Ad esempio, è possibile immettere un comando remoto per modificare una definizione di coda su un gestore code remoto.

Alcuni comandi non possono essere emessi in questo modo, in particolare, creando o avviando i gestori code e avviando i server dei comandi. Per eseguire questo tipo di attività, è necessario collegarsi al sistema remoto e immettere i comandi da tale ubicazione oppure creare un processo che possa emettere i comandi per conto dell'utente.

## **Gestione utilizzando i comandi MQSC su IBM i**

Utilizzare queste informazioni per informazioni sui comandi MQSC e su come utilizzarli per gestire IBM MQ for IBM i.

I comandi dello script IBM MQ (MQSC) sono scritti in formato leggibile, ossia in testo EBCDIC. Utilizzare i comandi MQSC per gestire gli oggetti gestore code, inclusi il gestore code stesso, le code, le definizioni di processo, gli elenchi nomi, i canali, i canali di connessione client, i listener, i servizi, gli argomenti e gli oggetti delle informazioni di autenticazione.

I comandi MQSC vengono immessi in un gestore code utilizzando il comando CL **STRMQMMQSC** IBM MQ . Questo metodo è solo un metodo batch, che prende il suo input da un file fisico di origine nel sistema della libreria del server. Il nome predefinito per questo file fisico origine è QMQSC.



**Attenzione:** Non utilizzare la libreria QTEMP come libreria di origine in STRMQMMQSC, poiché l'utilizzo della libreria QTEMP è limitato. È necessario utilizzare un'altra libreria come file di immissione per il comando.

IBM MQ for IBM i non fornisce un file di origine denominato QMQSC. Per elaborare i comandi MQSC, è necessario creare il file di origine QMQSC in una libreria di propria scelta, immettendo il seguente comando:

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT('IBM MQ - MQSC Source')
```

L'origine MQSC è contenuta nei membri all'interno di questo file di origine. Per gestire i membri, immettere il comando seguente:

```
WRKM BRPDM MYLIB/QMQSC
```

È ora possibile aggiungere nuovi membri e gestire quelli esistenti

È anche possibile immettere comandi MQSC in modo interattivo, immettendo RUNMQSC o:

1. Immettere il nome del gestore code e premere il tasto Enter per accedere al pannello dei risultati **WRKMQM**.
2. Selezionando F23=More options in questo pannello.
3. Selezionando l'opzione 26 rispetto a un gestore code attivo nel pannello mostrato in [Figura 23 a pagina 359](#).

Per terminare tale sessione MQSC, immettere end.

[Figura 23 a pagina 359](#) è un'estrazione da un file di comandi MQSC che mostra un comando MQSC (DEFINE QLOCAL) con relativi attributi.

```
.
.
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;
.
.
```

*Figura 23. Estrarre dal file di comandi MQSC, myprog.in*

Per la portabilità tra ambienti IBM MQ, limitare la lunghezza della riga nei file di comandi MQSC a 72 caratteri. Il segno più indica che il comando continua sulla riga successiva.

Gli attributi dell'oggetto specificati in MQSC vengono visualizzati in maiuscolo in questa sezione (ad esempio, RQMNAME), sebbene non siano sensibili al maiuscolo / minuscolo.

**Nota:**

1. Il formato di un file MQSC non dipende dalla sua ubicazione nel file system.
2. I nomi attributo MQSC sono limitati a otto caratteri.
3. I comandi di MQSC sono disponibili su altre piattaforme, incluso z/OS.

Per una descrizione di ciascun comando MQSC e della sua sintassi, consultare [Comandi MQSC](#).

## **Gestione mediante comandi PCF su IBM i**

Lo scopo dei comandi PCF ( IBM MQ programmable command format) è consentire la programmazione delle attività di gestione in un programma di amministrazione. In questo modo è possibile creare code e definizioni di processo e modificare gestori code da un programma.

I comandi PCF coprono la stessa gamma delle funzioni fornite dai comandi MQSC. Tuttavia, diversamente dai comandi MQSC, i comandi PCF e le risposte non sono in un formato di testo che è possibile leggere.

È possibile scrivere un programma per emettere comandi PCF per qualsiasi gestore code nella rete da un singolo nodo. In questo modo, è possibile centralizzare e automatizzare le attività di gestione.

Ogni comando PCF è una struttura di dati integrata nella parte di dati dell'applicazione di un messaggio IBM MQ. Ogni comando viene inviato al gestore code di destinazione utilizzando la funzione MQI MQPUT allo stesso modo di qualsiasi altro messaggio. Il server dei comandi sul gestore code che riceve il messaggio lo interpreta come messaggio di comandi ed esegue il comando. Per ottenere le repliche, l'applicazione emette una chiamata MQGET e i dati di risposta vengono restituiti in una struttura dati diversa. La domanda può quindi elaborare la risposta e agire di conseguenza.

Brevemente, queste sono alcune delle cose che il programmatore dell'applicazione deve specificare per creare un messaggio di comando PCF:

### Descrittore messaggio

Questo è un descrittore di messaggi IBM MQ standard, in cui:

- Il tipo di messaggio (*MsgType*) è MQMT\_REQUEST.
- Il formato del messaggio (*Format*) è MQFMT\_ADMIN.

### Dati applicazione

Contiene il messaggio PCF incluso l'intestazione PCF, in cui:

- Il tipo di messaggio PCF (*Type*) specifica MQCFT\_COMMAND.
- L'identificativo del comando specifica il comando, ad esempio *Change Queue* (MQCMD\_CHANGE\_Q).

I PCF di escape sono comandi PCF che contengono comandi MQSC all'interno del testo del messaggio. È possibile utilizzare i PCF per inviare comandi a un gestore code remoto. Consultare [“Utilizzo di MQAI per semplificare l'utilizzo dei PCF”](#) a pagina 36 per ulteriori informazioni.

Per una descrizione completa delle strutture dati PCF e come implementarle, consultare [Strutture per comandi e risposte](#).

## IBM i **Utilizzo di IBM MQ Explorer con IBM MQ for IBM i**

Utilizzare queste informazioni per gestire IBM MQ for IBM i utilizzando IBM MQ Explorer.

IBM MQ for Windows (piattaformax86) e IBM MQ per Linux (piattaformex86 e x86-64) forniscono un'interfaccia di gestione denominata Esplora risorse di IBM MQ per eseguire le attività di gestione come alternativa all'utilizzo dei comandi CL, di controllo o MQSC.

IBM MQ Explorer consente di eseguire la gestione locale o remota della propria rete da un computer su cui è in esecuzione Windows (piattaformax86) o Linux (piattaformex86 e x86-64), puntando IBM MQ Explorer ai gestori code e ai cluster a cui si è interessati.

Con IBM MQ Explorer, puoi:

- Avviare e arrestare un gestore code (solo sulla macchina locale).
- Definire, visualizzare e modificare le definizioni degli oggetti IBM MQ come code, argomenti e canali.
- Esaminare i messaggi su una coda.
- Avviare e arrestare un canale.
- Visualizzare le informazioni di stato su un canale.
- Visualizzare i gestori code in un cluster.
- Verificare quali applicazioni, utenti o canali hanno una particolare coda aperta.
- Creare un nuovo cluster di gestori code utilizzando la procedura guidata **Crea nuovo cluster**.
- Aggiungere un gestore code a un cluster utilizzando il wizard **Aggiungi gestore code al cluster**.
- Gestire l'oggetto delle informazioni di autenticazione, utilizzato con la sicurezza del canale TLS (Transport Layer Security).

Utilizzando la guida online, è possibile:

- Definire e controllare varie risorse, inclusi gestori code, code, canali, definizioni di processi, canali di connessione client, listener, argomenti, servizi, elenchi nomi e cluster.
- Avviare o arrestare un gestore code e i relativi processi associati.
- Visualizzare i gestori code e i relativi oggetti associati sulla stazione di lavoro o da altre stazioni di lavoro.
- Verificare lo stato dei gestori code, dei cluster e dei canali.

Accertarsi di aver soddisfatto i requisiti riportati di seguito prima di tentare di utilizzare IBM MQ Explorer per gestire IBM MQ su una macchina server. Verificare che:

1. Un server dei comandi è in esecuzione per qualsiasi gestore code gestito, avviato sul server dal comando CL **STRMQMSVR**.
2. Esiste un listener TCP/IP adatto per ogni gestore code remoto. Questo è il listener IBM MQ avviato dal comando **STRMQMLSR**.
3. Il canale di connessione server, denominato `SYSTEM.ADMIN.SVRCONN`, esiste su ogni gestore code remoto. È necessario creare questo canale. È obbligatorio per ogni gestore code remoto gestito. Senza di esso, la gestione remota non è possibile.
4. Verificare che la coda `SYSTEM.MQEXPLORER.REPLY.MODEL` esista.

## **Gestione del server dei comandi per la gestione in remoto su IBM i**

Utilizzare queste informazioni per informazioni sulla gestione remota di IBM MQ for IBM i Command Server.

Ciascun gestore code può avere un server dei comandi associato. Un server dei comandi elabora i comandi in entrata dai gestori code remoti o i comandi PCF dalle applicazioni. Presenta i comandi al gestore code per l'elaborazione e restituisce un codice di completamento o un messaggio dell'operatore in base all'origine del comando.

Un server dei comandi è obbligatorio per tutte le attività di gestione che coinvolgono PCF, MQAI e anche per la gestione remota.

**Nota:** Per la gestione remota, è necessario assicurarsi che il gestore code di destinazione sia in esecuzione. Altrimenti, i messaggi che contengono i comandi non possono lasciare il gestore code da cui vengono emessi. Invece, questi messaggi vengono accodati nella coda di trasmissione locale che serve il gestore code remoto. Evitare questa situazione, se possibile.

Esistono comandi di controllo separati per avviare e arrestare il server dei comandi. È possibile eseguire le operazioni descritte nelle sezioni seguenti utilizzando Esplora risorse di IBM MQ.

### **Avvio e arresto del server dei comandi**

Per avviare il server dei comandi, utilizzare questo comando CL:

```
STRMQMSVR MQMNAME('saturn.queue.manager')
```

dove `saturn.queue.manager` è il gestore code per cui viene avviato il server dei comandi.

Per arrestare il server dei comandi, utilizzare uno dei comandi CL riportati di seguito:

1. 

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRLD)
```

per eseguire un arresto controllato, dove `saturn.queue.manager` è il gestore code per cui si sta arrestando il server dei comandi. Questa è l'opzione predefinita, che significa che `OPTION(*CNTRLD)` può essere omesso.

2. 

```
ENDMQMSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)
```

per eseguire un arresto immediato, dove `saturn.queue.manager` è il gestore code per cui viene arrestato il server dei comandi.

### **Visualizzazione dello stato del server dei comandi**

Per la gestione remota, verificare che sia in esecuzione il server dei comandi sul gestore code di destinazione. Se non è in esecuzione, i comandi remoti non possono essere elaborati. Tutti i messaggi che contengono i comandi vengono accodati nella coda comandi del gestore code di destinazione `SYSTEM.ADMIN.COMMAND.QUEUE`.

Per visualizzare lo stato del server dei comandi per un gestore code, denominato `saturn.queue.manager`, il comando CL è:

```
DSPMQMSVR MQMNAME('saturn.queue.manager')
```

Immettere questo comando sulla macchina di destinazione. Se il server dei comandi è in esecuzione, viene visualizzato il pannello mostrato in [Figura 24 a pagina 362](#) :

```
Display MQM Command Server (DSPMQMSVR)
```

```
Queue manager name . . . . . > saturn.queue.manager
```

```
MQM Command Server Status. . . . > RUNNING
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys
```

*Figura 24. Visualizza pannello MQM Command Server*

## Esecuzione dei comandi della console Web

Devi configurare il tuo ambiente come descritto nel seguente testo, affinché i comandi Qshell correlati alla console web vengano eseguiti correttamente su IBM MQ for IBM i.

### Informazioni su questa attività

Quando Qshell viene avviato, inizializza le tabelle interne per l'elaborazione dei comandi basati sul CCSID del lavoro. Affinché i comandi Qshell relativi alla console Web vengano eseguiti correttamente su IBM i, devi configurare il tuo ambiente.

Una locale viene impostata impostando la variabile di ambiente LANG sul nome percorso di un oggetto locale. Ad esempio, per impostare la locale per l'inglese americano, la variabile di ambiente LANG è impostata come segue:

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

Nella Qshell, è possibile controllare l'impostazione immettendo la serie di comandi per elencare le variabili di ambiente. Di solito è LANG che può influire sulla locale per l'ambiente di runtime. Può anche avere LC\_ALL.

Per eseguire correttamente i comandi Qshell, l'impostazione dell'ambiente locale deve essere congruente con l'impostazione del lavoro.

### Procedura

Utilizzare il comando CL DSPJOB JOB (JobNumber/USERProfile/JobName)

- a) Selezionare l'opzione 2 per visualizzare gli attributi di definizione lavoro.
- b) I seguenti attributi devono essere congruenti con l'impostazione dell'ambiente LANG o LC\_ALL
  - Identificativo lingua
  - Identificativo paese o regione
  - CCSID (Coded character set identifier)

Ad esempio, se

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

gli attributi del lavoro devono essere:

- ID lingua..... FRA
- ID paese o regione..... FR
- CCSID (Coded character set ID).. 297

## Operazioni successive

Per ulteriori informazioni sul supporto lingua nazionale, consultare l'argomento [IBM Documentation Considerazioni sul supporto lingua nazionale \(NLS\)](#).

## IBM i Gestione del lavoro per IBM i

Queste informazioni descrivono il modo in cui IBM MQ gestisce le richieste di lavoro e descrivono le opzioni disponibili per assegnare una priorità e controllare i lavori associati a IBM MQ.

### Avviso

Non modificare gli oggetti di gestione lavoro IBM MQ a meno che non si comprendano completamente i concetti di gestione lavoro IBM i e IBM MQ .

Ulteriori informazioni sui sottosistemi e le descrizioni dei lavori sono disponibili in [Gestione lavoro](#) nella documentazione del prodotto IBM i . Prestare particolare attenzione alle sezioni [Avvio dei lavori](#) e [Lavori batch](#).

IBM MQ for IBM i integra l'ambiente IBM i UNIX e i thread IBM i . **Non** apportare alcuna modifica agli oggetti in IFS ( Integrated File System ) .

Durante le normali operazioni, un gestore code IBM MQ avvia una serie di lavori batch per eseguire attività differenti. Per impostazione predefinita questi lavori batch vengono eseguiti nel sottosistema QMQM creato quando IBM MQ è installato.

La gestione del lavoro si riferisce al processo di personalizzazione delle attività IBM MQ per ottenere le prestazioni ottimali dal sistema o per semplificare l'amministrazione.

Ad esempio, puoi:

- Modificare la priorità di esecuzione dei lavori per rendere un gestore code più reattivo di un altro.
- Reindirizzare l'output di un numero di lavori ad una particolare coda di emissione.
- Eseguire tutti i lavori di un determinato tipo in uno specifico sottosistema.
- Isolare gli errori in un sottosistema.

La gestione lavoro viene eseguita creando o modificando le descrizioni lavoro associate ai lavori IBM MQ . È possibile configurare la gestione del lavoro per:

- Un'intera installazione di IBM MQ .
- Singoli gestori code.
- Singoli lavori per singoli gestori code.

## IBM i Attività IBM MQ per IBM i

Questa è una tabella dei job IBM MQ for IBM i e una breve descrizione di ognuno.

Quando un gestore code è in esecuzione, vengono visualizzati alcuni o tutti i seguenti lavori batch in esecuzione nel profilo utente QMQM nel sottosistema IBM MQ . I lavori sono descritti brevemente in [Tabella 21 a pagina 364](#).

È possibile visualizzare tutti i lavori connessi ad un gestore code utilizzando l'opzione 22 nel pannello WRKMQM ( **Gestione gestore code** ). È possibile visualizzare i listener utilizzando il comando WRKMQMLSR.

*Tabella 21. Attività IBM MQ .*

Nome lavoro	Funzione
AMQZMUCO	Gestore programmi di utilità. Questo lavoro esegue i programmi di utilità del gestore code critici, ad esempio il gestore della catena journal.
AMQZXMAO	Il controller di esecuzione che è il primo lavoro avviato dal gestore code. Gestisce le richieste MQCONN e avvia i processi dell'agent per elaborare le chiamate API IBM MQ .
AMQZFUMA	OAM (Object authority manager).
AMQZLAAO	Gli agent del gestore code che eseguono la maggior parte del lavoro per le applicazioni che si collegano al gestore code utilizzando MQCNO_STANDARD_BINDING.
AMQZLSAO	Agent gestore code.
AMQZMUFO	Gestore utilità
AMQZMGRO	Controller processi. Questo lavoro viene utilizzato per avviare e gestire listener e servizi.
AMQZMURO	Gestore programmi di utilità. Questo lavoro esegue i programmi di utilità del gestore code critici, ad esempio il gestore della catena journal.
AMQFQPUB	Daemon di pubblicazione / sottoscrizione accodato.
AMQFCXBA	Lavoro di lavoro del broker.
RUNMQBRK	Lavoro di controllo broker.
AMQRMPPA	Lavoro di pool del processo del canale.
AMQCRSTA	Risponditore del canale richiamato da TCP/IP.
AMQCRS6B	LU62 canale ricevente e connessione client (vedere nota).
AMQRRMFA	Gestore repository per i cluster.
AMQCLMAA	Listener TCP/IP non in thread.
AMQPCSEA	Il processore comandi PCF che gestisce le richieste di gestione remota e PCF.
RUNMQTRM	Controllo trigger.
RUNMQDLQ	Gestore code di messaggi non instradabili.
RUNMQCHI	L'iniziatore del canale.
RUNMQCHL	Lavoro del canale mittente avviato per ogni canale mittente.
RUNMQLSR	Listener TCP/IP con thread.
AMQRCMLA	Canale MQSC e processore comandi PCF.

**Nota:** Il lavoro ricevitore LU62 viene eseguito nel sottosistema di comunicazione e acquisisce le proprietà di runtime dalle voci di instradamento e di comunicazione utilizzate per avviare il lavoro. Per ulteriori informazioni, consultare [Fine avviata \(Ricevitore\)](#) .

## **Oggetti di gestione lavoro su IBM i**

Quando si installa IBM MQ , vengono forniti diversi oggetti nella libreria QMQM per assistere nella gestione del lavoro. Questi oggetti sono quelli necessari per l'esecuzione dei lavori IBM MQ nel relativo sottosistema.

Le descrizioni dei lavori di esempio vengono fornite per due dei lavori batch IBM MQ . Se non viene fornita alcuna descrizione lavoro specifica per un lavoro IBM MQ , viene eseguito con la descrizione lavoro predefinita QMQMJOB.

Gli oggetti di gestione del lavoro forniti quando si installa IBM MQ sono elencati in [Tabella 22 a pagina 365](#) e gli oggetti creati per un gestore code sono elencati in [Tabella 23 a pagina 365](#)

**Nota:** È possibile trovare gli oggetti di gestione del lavoro nella libreria QMQM e gli oggetti del gestore code nella libreria del gestore code.

*Tabella 22. Oggetti di gestione lavoro*

Nome	Tipo	Descrizione
AMQZLAA0	*JOB	La descrizione lavoro utilizzata dai processi dell'agente IBM MQ
AMQZLSA0	*JOB	L'agent del gestore code dei bind isolati
AMQZXMA0	*JOB	La descrizione lavoro utilizzata dai controller di esecuzione IBM MQ
QMQM	*SBS	Il sottosistema in cui vengono eseguiti tutti i lavori IBM MQ
QMQM	*JOB	La coda lavori collegata al sottosistema fornito
QMQMJOB	*JOB	La descrizione lavoro IBM MQ predefinita, utilizzata se non esiste una descrizione lavoro specifica per un lavoro
MQMMSG	*MSG	La coda messaggi predefinita per i lavori IBM MQ .
QMQMRUN20	*CLS	Una descrizione di classe per lavori IBM MQ ad alta priorità
QMQMRUN35	*CLS	Una descrizione di classe per lavori IBM MQ con priorità media
QMQMRUN50	*CLS	Una descrizione della classe per lavori IBM MQ a bassa priorità

*Tabella 23. Oggetti di gestione lavoro creati per un gestore code*

Nome	Tipo	Descrizione
AMQA000000	*JRNRCV	Ricevitore giornale locale
JRN AMQA	*JRN	Giornale locale
RNINF AMQJR	*USRSPC	Lo spazio utente aggiornato con gli ultimi ricevitori di giornale richiesti per l'avvio e il ripristino dei supporti di un gestore code. Questo spazio utente può essere interrogato da un'applicazione per determinare quali ricevitori di giornale richiedono l'archiviazione e quali possono essere cancellati in modo sicuro.
MQAJRNMSG	*MSG	Coda messaggi giornale locale
AMQCRC6B	*PGM	Programma per avviare la connessione LU6.2
AMQRFOLD	*FILE	File di definizione del canale del gestore code migrato
MQMMSG	*MSG	Coda messaggi del gestore code

## Come IBM MQ utilizza gli oggetti di gestione del lavoro su IBM i

Queste informazioni descrivono il modo in cui IBM MQ utilizza gli oggetti di gestione lavoro e forniscono esempi di configurazione.



**Attenzione:** Non modificare le impostazioni della voce della coda lavori nel sottosistema QMQM per limitare il numero di lavori consentiti nel sottosistema in base alla priorità. Se si tenta di eseguire questa operazione, è possibile arrestare l'esecuzione dei lavori IBM MQ essenziali dopo che sono stati inoltrati e causare l'errore di avvio del gestore code.

Per comprendere come configurare la gestione del lavoro, è necessario prima comprendere come IBM MQ utilizza le descrizioni del lavoro.

La descrizione lavoro utilizzata per avviare il lavoro controlla molti attributi del lavoro. Ad esempio:

- La coda lavori su cui il lavoro è accodato e su cui viene eseguito il sottosistema.
- I dati di instradamento utilizzati per avviare il lavoro e la classe che il lavoro utilizza per i parametri di runtime.
- La coda di emissione utilizzata dal lavoro per i file di stampa.

Il processo di avvio di un lavoro IBM MQ può essere considerato in tre fasi:

1. IBM MQ seleziona una descrizione lavoro.

IBM MQ utilizza la seguente tecnica per determinare quale descrizione lavoro utilizzare per un lavoro batch:

- a. Cercare nella libreria del gestore code una descrizione lavoro con lo stesso nome del lavoro. Per ulteriori dettagli sulla libreria dei gestori code, consultare la sezione [Informazioni sui nomi delle librerie dei gestori code IBM MQ for IBM i](#).
- b. Cercare nella libreria del gestore code la descrizione lavoro predefinita QMQMJOB.
- c. Cercare nella libreria QMQM una descrizione lavoro con lo stesso nome del lavoro.
- d. Utilizzare la descrizione lavoro predefinita, QMQMJOB, nella libreria QMQM.

2. Il lavoro viene inoltrato alla coda lavori.

Le descrizioni dei lavori fornite con IBM MQ sono state impostate, per impostazione predefinita, per inserire i lavori nella coda lavori QMQM nella libreria QMQM. La coda lavori QMQM è collegata al sottosistema QMQM fornito, quindi per impostazione predefinita i lavori vengono avviati nel sottosistema QMQM.

3. Il lavoro entra nel sottosistema e passa attraverso le fasi di instradamento.

Quando il lavoro entra nel sottosistema, i dati di instradamento specificati nella descrizione lavoro vengono utilizzati per trovare le voci di instradamento per il lavoro.

I dati di instradamento devono corrispondere a una delle voci di instradamento definite nel sistema secondario QMQM e ciò definisce quale delle classi fornite (QMQMRUN20, QMQMRUN35 o QMQMRUN50) viene utilizzata dal lavoro.

**Nota:** Se i lavori IBM MQ non sembrano essere in fase di avvio, assicurarsi che il sistema secondario sia in esecuzione e che la coda lavori non sia congelata,

Se sono stati modificati gli oggetti di gestione lavoro IBM MQ, assicurarsi che tutto sia associato correttamente. Ad esempio, se si specifica una coda lavori diversa da QMQM/QMQM nella descrizione del lavoro, assicurarsi che ADDJOBQE venga eseguito per il sottosistema, ossia QMQM.

È possibile creare una descrizione lavoro per ogni lavoro documentato in [Tabella 21 a pagina 364](#) utilizzando il seguente foglio di lavoro come esempio:

```
What is the queue manager library name? _____
Does job description AMQZXMA0 exist in the queue manager library? Yes No
Does job description QMQMJOB exist in the queue manager library? Yes No
Does job description AMQZXMA0 exist in the QMQM library? Yes No
Does job description QMQMJOB exist in the QMQM library? Yes No
```

Se si risponde No a tutte queste domande, creare una descrizione del lavoro globale QMQMJOB nella libreria QMQM.

## La coda messaggi IBM MQ

Una coda messaggi IBM MQ , QMQMMSG, viene creata in ogni libreria del gestore code. I messaggi del sistema operativo vengono inviati a questa coda alla fine dei lavori del gestore code e IBM MQ invia i messaggi alla coda. Ad esempio, per notificare quali ricevitori di giornale sono necessari all'avvio. Mantenere il numero di messaggi in questa coda messaggi ad una dimensione gestibile per semplificare il monitoraggio.

### IBM i Esempi di sistema predefiniti per IBM i

Questi esempi mostrano come funziona un'installazione di IBM MQ non modificata quando alcuni dei lavori standard vengono inoltrati all'avvio del gestore code.

Innanzitutto, viene avviato il lavoro del controller di esecuzione AMQZXMA0 .

1. Immettere il comando **STRMQM** per il gestore code TESTQM.
2. IBM MQ ricerca la libreria del gestore code QMTESTQM, prima per la descrizione lavoro AMQZXMA0, e quindi la descrizione lavoro QMQMJOB.

Nessuna di queste descrizioni lavoro esiste, quindi IBM MQ ricerca la descrizione lavoro AMQZXMA0 nella libreria del prodotto QMQM. Questa descrizione lavoro esiste, quindi viene utilizzata per inoltrare il lavoro.

3. La descrizione del lavoro utilizza la coda lavori predefinita IBM MQ , quindi il lavoro viene inoltrato alla coda lavori QMQM/QMQM.
4. I dati di instradamento sulla descrizione del lavoro AMQZXMA0 è QMQMRUN20, quindi il sistema ricerca le voci di instradamento del sottosistema per una che corrisponda a tali dati.

Per impostazione predefinita, la voce di instradamento con numero di sequenza 9900 ha dati di confronto che corrispondono a QMQMRUN20, quindi il lavoro viene avviato con la classe definita su tale voce di instradamento, denominata anche QMQMRUN20.

5. La classe QMQM/QMQMRUN20 ha la priorità di esecuzione impostata su 20, quindi il lavoro AMQZXMA0 viene eseguito nel sottosistema QMQM con la stessa priorità della maggior parte dei lavori interattivi sul sistema.

### IBM i Configurazione degli esempi di gestione lavoro su IBM i

Utilizzare queste informazioni per informazioni su come modificare e creare descrizioni lavoro IBM MQ per modificare gli attributi di runtime dei lavori IBM MQ .

La chiave per la flessibilità della gestione lavoro IBM MQ risiede nel modo a due livelli in cui IBM MQ ricerca le descrizioni dei lavori:

- Se si creano o si modificano le descrizioni lavoro in una libreria del gestore code, tali modifiche sovrascrivono le descrizioni lavoro globali in QMQM, ma le modifiche sono locali e influiscono solo su quel particolare gestore code.
- Se si creano o si modificano le descrizioni lavoro globali nella libreria QMQM, tali descrizioni lavoro influiscono su tutti i gestori code sul sistema, a meno che non vengano sovrascritte localmente per i singoli gestori code.

1. Il seguente esempio aumenta la priorità dei lavori di controllo canale per un singolo gestore code.

Per fare in modo che i lavori del gestore repository e dell'iniziatore del canale, AMQRRMFA e RUNMQCHI, vengano eseguiti il più rapidamente possibile per il gestore code TESTQM, effettuare le seguenti operazioni:

- a. Creare i duplicati locali della descrizione del lavoro QMQM/QMQMJOB con i nomi dei processi IBM MQ che si desidera controllare nella libreria del gestore code. Ad esempio:

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
NEWOBJ (RUNMQCHI)
```

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
NEWOBJ (AMQRRMFA)
```

- b. Modificare il parametro dei dati di instradamento nella descrizione lavoro per assicurarsi che i lavori utilizzino la classe QMQMRUN20 .

```
CHGJOB JOB(QMTESTQM/RUNMQCHI) RTGDTA('QMQMRUN20')
CHGJOB JOB(QMTESTQM/AMQRRMFA) RTGDTA('QMQMRUN20')
```

I lavori AMQRRMFA e RUNMQCHI per il gestore code TESTQM ora:

- Utilizzare le nuove descrizioni dei lavori locali nella libreria del gestore code
  - Eseguire con priorità 20, poiché la classe QMQMRUN20 viene utilizzata quando i lavori entrano nel sottosistema.
2. Il seguente esempio definisce una nuova classe di priorità di esecuzione per il sottosistema QMQM.
- a. Creare una classe duplicata nella libreria QMQM, per consentire ad altri gestori code di accedere alla classe, immettendo il seguente comando:

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ (QMQMRUN10)
```

- b. Modificare la classe in modo che abbia la nuova priorità di esecuzione immettendo il seguente comando:

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. Aggiungere la nuova definizione di classe al sottosistema immettendo il comando seguente:

```
ADDRTGE SBS(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)
CLS(QMQM/QMQMRUN10)
```

**Nota:** È possibile specificare un qualsiasi valore numerico per il numero di sequenza di instradamento, ma i valori devono essere in ordine sequenziale. Questo numero di sequenza indica al sottosistema l'ordine in cui le voci di instradamento devono essere ricercate per una corrispondenza dei dati di instradamento.

- d. Modificare la descrizione del lavoro locale o globale per utilizzare la nuova classe di priorità immettendo il seguente comando:

```
CHGJOB JOB(QMQMLibname/QMQMJOB) RTGDTA('QMQMRUN10')
```

Ora tutti i lavori del gestore code associati al QMLibraryname utilizzano una priorità di esecuzione di 10.

3. Il seguente esempio esegue un gestore code nel proprio sottosistema

Per eseguire tutti i lavori per il gestore code TESTQM nel sottosistema QBATCH, effettuare le seguenti operazioni:

- a. Creare un duplicato locale della descrizione lavoro QMQM/QMQMJOB nella libreria del gestore code con il comando

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMTESTQM)
```

- b. Modificare il parametro della coda lavori nella descrizione lavoro per assicurarsi che i lavori utilizzino la coda lavori QBATCH.

```
CHGJOB JOB(QMTESTQM/QMQMJOB) JOBQ(*LIBL/QBATCH)
```

**Nota:** La coda lavori è associata alla descrizione del sottosistema. Se si rileva che i lavori si trovano sulla coda lavori, verificare che la definizione della coda lavori sia definita su SBSB. Utilizzare il comando DSPSBSB per il sistema secondario e scegliere l'opzione 6, Voci coda lavori.

Tutti i lavori per il gestore code TESTQM ora:

- Utilizzare la nuova descrizione del lavoro predefinito locale nella libreria del gestore code
- Vengono inoltrati alla coda lavori QBATCH.

Per garantire che i lavori vengano instradati e impostati correttamente le priorità:

- Creare le voci di instradamento per i lavori IBM MQ nel sottosistema QBATCH oppure
- Basarsi su una voce di instradamento catch-all che richiama QCMD, indipendentemente dai dati di instradamento utilizzati.

Questa opzione funziona solo se l'opzione del numero massimo di lavori attivi per la coda lavori QBATCH è impostato su \*NOMAX. Il valore predefinito del sistema è 1.

#### 4. Il seguente esempio crea un altro sottosistema IBM MQ

- a. Creare un sottosistema duplicato nella libreria QMQM immettendo il seguente comando:

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBSB) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. Rimuovere la coda lavori QMQM immettendo il seguente comando:

```
RMVJOBQE SBSB(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. Creare una nuova coda lavori per il sottosistema immettendo il seguente comando:

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. Aggiungere una voce coda lavori al sottosistema immettendo il comando seguente:

```
ADDJOBQE SBSB(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. Creare un QMQMJOB duplicato nella libreria del gestore code immettendo il seguente comando:

```
CRTDUPOBJ OBJ(QMQMJOB) FROMLIB(QMQM) OBJTYPE(*JOB) TOLIB(QMlibraryname)
```

- f. Modificare la descrizione lavoro per utilizzare la nuova coda lavori immettendo il seguente comando:

```
CHGJOB JOB(QMlibraryname/QMQMJOB) JOBQ(QMQM/QMQM2)
```

- g. Avviare il sottosistema immettendo il seguente comando:

```
STRSBS SBSB(QMQM/QMQM2)
```

#### **Nota:**

- È possibile specificare il sottosistema in qualsiasi libreria. Se per qualsiasi motivo il prodotto viene reinstallato o la libreria QMQM viene sostituita, tutte le modifiche apportate vengono rimosse.
- Tutti i lavori del gestore code associati al QMlibraryname vengono ora eseguiti nel sottosistema QMQM2.

## Disponibilità, backup, ripristino e riavvio su IBM i

Utilizzare queste informazioni per comprendere in che modo IBM MQ for IBM i utilizza il supporto journal IBM i per facilitare la propria strategia di backup e ripristino.

Prima di leggere questa sezione, è necessario avere dimestichezza con i metodi di backup e ripristino standard di IBM i e con l'utilizzo dei giornali e dei relativi ricevitori di giornale associati su IBM i. Per informazioni su questi argomenti, consultare [Backup e ripristino](#).

Per comprendere la strategia di backup e ripristino, è necessario prima comprendere il modo in cui IBM MQ for IBM i organizza i propri dati nel file system IBM i e nell'IFS (integrated file system).

IBM MQ for IBM i conserva i dati in una singola libreria per ciascuna istanza del gestore code e nei file di flusso nel file system IFS.

Le librerie specifiche del gestore code contengono i journal, i ricevitori del journal e gli oggetti richiesti per controllare la gestione del lavoro del gestore code. I file e gli indirizzari IFS contengono i file di configurazione IBM MQ, le descrizioni degli oggetti IBM MQ e i dati in essi contenuti.

Ogni modifica a questi oggetti, ripristinabile in un errore di sistema, viene registrata in un giornale *prima* che venga applicata all'oggetto appropriato. Ciò ha l'effetto che tali modifiche possono essere recuperate riproducendo le informazioni registrate nel giornale.

È possibile configurare IBM MQ for IBM i per utilizzare più istanze del gestore code su server differenti per fornire una maggiore disponibilità del gestore code e accelerare il ripristino in caso di errore del server o del gestore code.

## Journal del gestore code su IBM i

Utilizzare queste informazioni per comprendere come IBM MQ for IBM i utilizza i giornali nelle proprie operazioni per controllare gli aggiornamenti agli oggetti locali.

Ogni libreria del gestore code contiene un journal per tale gestore code e il journal ha il nome `QM GRLIB/AMQ A JRN`, dove `QM GRLIB` è il nome della libreria del gestore code e `A` è una lettera, `A` nel caso di un gestore code a istanza singola, che è univoco per l'istanza del gestore code.

`QM GRLIB` prende il nome `QM`, seguito dal nome del gestore code in un formato univoco. Ad esempio, un gestore code denominato `TEST` ha una libreria di gestori code denominata `QMTEST`. La libreria del gestore code può essere specificata quando si crea un gestore code utilizzando il comando **CRTMQM**.

I giornali hanno ricevitori di giornale associati che contengono le informazioni che si stanno registrando su giornale. I ricevitori sono oggetti a cui le informazioni possono essere solo accodate e si riempiranno alla fine.

I ricevitori di giornale utilizzano spazio su disco prezioso con informazioni non aggiornate. Tuttavia, è possibile inserire le informazioni nella memoria permanente per ridurre al minimo questo problema. Un ricevitore di giornale è collegato al giornale in qualsiasi momento. Se il ricevitore di giornale raggiunge la sua dimensione di soglia predeterminata, viene scollegato e sostituito da un nuovo ricevitore di giornale. È possibile specificare la soglia dei ricevitori del journal quando si crea un gestore code utilizzando **CRTMQM** e il parametro **THRESHOLD**.

I ricevitori del journal associati al journal locale IBM MQ per IBM i esistono in ogni libreria del gestore code e adottano una convenzione di denominazione come segue:

```
AMQ Arnnnnn
```

dove

**A**

è una lettera A-Z. È A per i gestori code a istanza singola. Varia per le diverse istanze di un gestore code a più istanze.

**nnnnn**

è decimale 00000 to 99999 che viene incrementato di 1 per il giornale successivo nella sequenza.

*r*

è decimale 0 to 9, che viene incrementato di 1 ogni volta che un ricevitore viene ripristinato.

La sequenza dei journal si basa sulla data. Tuttavia, la denominazione del giornale successivo si basa sulle regole seguenti:

1. AMQA $r$ nnnnn passa a AMQA $r$ (nnnnn+1) e nnnnn esegue il wrapping quando raggiunge 99999. Ad esempio, AMQA099999 va in AMQA000000e AMQA999999 va in AMQA900000.
2. Se già esiste un giornale con un nome generato dalla regola 1, il messaggio CPI70E3 viene inviato alla coda messaggi QSYSOPR e la commutazione automatica del ricevitore si arresta.

Il ricevitore attualmente collegato continua ad essere utilizzato fino a quando non si esamina il problema e si collega manualmente un nuovo ricevitore.

3. Se non è disponibile alcun nuovo nome nella sequenza (ovvero, tutti i nomi di giornale possibili si trovano sul sistema), è necessario effettuare entrambe le seguenti operazioni:
  - a. Eliminare i giornali non più necessari (consultare [“Gestione giornale su IBM i”](#) a pagina 375 ).
  - b. Registrare le modifiche di giornale nell'ultimo ricevitore di giornale utilizzando ( **RCDMQMIMG** ) e quindi ripetere il passo precedente. Ciò consente il riutilizzo dei vecchi nomi dei ricevitori di giornale.

Il giornale AMQAJRN utilizza l'opzione MNGRCV (\*SYSTEM) per consentire al sistema operativo di modificare automaticamente i ricevitori di giornale quando viene raggiunta la soglia. Per ulteriori informazioni su come il sistema gestisce i ricevitori, consultare *IBM i Backup e ripristino*.

Il valore di soglia predefinito del ricevitore di giornale è 100.000 KB. È possibile impostare questo valore su un valore maggiore quando si crea il gestore code. Il valore iniziale dell'attributo LogReceiverDimensione viene scritto nella stanza LogDefaults del file mqs.ini.

Quando un ricevitore di giornale supera la soglia specificata, il ricevitore viene scollegato e viene creato un nuovo ricevitore di giornale, ereditando gli attributi dal precedente ricevitore. Le modifiche apportate agli attributi LogReceiverSize o LogASP dopo la creazione di un gestore code vengono ignorate quando il sistema collega automaticamente un nuovo ricevitore di giornale

Consultare [Modifica delle informazioni di configurazione su IBM i](#) per ulteriori dettagli sulla configurazione del sistema.

Se è necessario modificare la dimensione dei ricevitori del giornale dopo che il gestore code è stato creato, creare un nuovo ricevitore del giornale e impostarne il proprietario su QMQM utilizzando i seguenti comandi:

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +  
TEXT('MQM LOCAL JOURNAL RECEIVER')  
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

dove

### **QMGLIB**

È il nome della libreria del gestore code

### **A**

È l'identificativo dell'istanza (di norma A).

### **rrnnnnn**

Indica il successivo ricevitore di giornale nella sequenza di denominazione descritta precedentemente

### **xxxxxx**

Indica la nuova soglia del ricevitore (in KB)

**Nota:** La dimensione massima del ricevitore è regolata dal sistema operativo. Per verificare questo valore, consultare la parola chiave THRESHOLD sul comando **CRTJRNRCV**.

Ora collegare il nuovo ricevitore al giornale AMQAJRN con il comando:

```
CHGJRN JRN(QMGLIB/AMQ A JRN) JRNRCV(QMGLIB/AMQ Annnnnn)
```

Consultare [“Gestione giornale su IBM i”](#) a pagina 375 per i dettagli su come gestire questi ricevitori di giornale.

### IBM i **Utilizzo del journal del gestore code su IBM i**

Utilizzare queste informazioni per comprendere in che modo IBM MQ for IBM i utilizza i giornali nelle proprie operazioni per controllare gli aggiornamenti agli oggetti locali.

Gli aggiornamenti permanenti alle code messaggi si verificano in due fasi. I record che rappresentano l'aggiornamento vengono prima scritti nel giornale, quindi il file di coda viene aggiornato.

I ricevitori di giornale possono quindi diventare più aggiornati dei file della coda. Per garantire che l'elaborazione del riavvio inizi da un punto congruente, IBM MQ utilizza i punti di controllo.

Un punto di controllo è un momento in cui il record descritto nel giornale è uguale al record nella coda. Il punto di controllo stesso è costituito dalla serie di record journal necessari per riavviare il gestore code. Ad esempio, lo stato di tutte le transazioni (ossia, le unità di lavoro) attive al momento del punto di controllo.

I punti di controllo vengono generati automaticamente da IBM MQ. Vengono presi quando il gestore code viene avviato e arrestato e dopo un certo numero di operazioni vengono registrate.

È possibile forzare un gestore code a prendere un punto di controllo immettendo il comando RCDMQMIMG su tutti gli oggetti su un gestore code e visualizzando i seguenti risultati:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNTA(*YES)
```

Man mano che le code gestiscono ulteriori messaggi, il record del punto di controllo diventa incongruente con lo stato corrente delle code.

Quando IBM MQ viene riavviato, individua l'ultimo record del punto di controllo nel log. Queste informazioni vengono conservate nel file del punto di controllo aggiornato alla fine di ogni punto di controllo. Il record del punto di controllo rappresenta il punto di coerenza più recente tra il log e i dati. I dati da questo punto di controllo vengono utilizzati per ricreare le code esistenti al momento del punto di controllo. Quando le code vengono ricreate, il log viene riprodotto in avanti per riportare le code allo stato in cui si trovavano prima dell'errore di sistema o della chiusura.

Per comprendere in che modo IBM MQ utilizza il journal, considerare il caso di una coda locale denominata TESTQ nel gestore code TEST. Questo è rappresentato dal file IFS:

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

Se un messaggio specificato viene inserito in questa coda e quindi richiamato dalla coda, le azioni che si verificano vengono mostrate nella figura [Figura 25 a pagina 372](#).

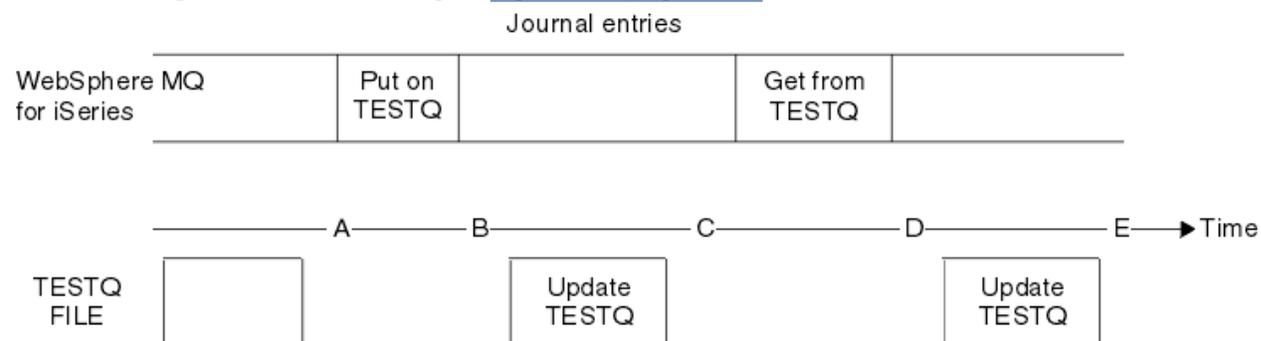


Figura 25. Sequenza di eventi durante l'aggiornamento degli oggetti MQM

I cinque punti, da A a E, mostrati nel diagramma rappresentano i punti nel tempo che definiscono i seguenti stati:

- A** La rappresentazione del file IFS della coda è congruente con le informazioni contenute nel giornale.
- B** Una voce di giornale viene scritta nel giornale che definisce un'operazione Put sulla coda.
- C** L'aggiornamento appropriato viene effettuato sulla coda.
- D** Una voce di giornale viene scritta nel giornale che definisce un'operazione Get dalla coda.
- E** L'aggiornamento appropriato viene effettuato sulla coda.

La chiave per le funzioni di recupero di IBM MQ for IBM i è che l'utente può salvare la rappresentazione del file IFS di TESTQ come all'ora A e successivamente ripristinare la rappresentazione del file IFS di TESTQ come all'ora E, ripristinando l'oggetto salvato e riproducendo le voci nel giornale dall'ora A in poi.

Questa strategia viene utilizzata da IBM MQ for IBM i per ripristinare i messaggi persistenti dopo un malfunzionamento del sistema. IBM MQ ricorda una particolare voce nei ricevitori del giornale e garantisce che all'avvio riproduce le voci nei giornali da questo punto in poi. Questa voce di avvio viene periodicamente ricalcolata in modo che IBM MQ debba eseguire solo la ripetizione minima necessaria al successivo avvio.

IBM MQ fornisce il ripristino individuale degli oggetti. Tutte le informazioni persistenti relative ad un oggetto vengono registrate nei giornali IBM MQ for IBM i locali. Qualsiasi oggetto IBM MQ danneggiato o danneggiato può essere completamente ricostruito dalle informazioni contenute nel giornale.

Per ulteriori informazioni su come il sistema gestisce i ricevitori, consultare [“Disponibilità, backup, ripristino e riavvio su IBM i”](#) a pagina 370.

### **Immagini multimediali su IBM i**

Su IBM i, un'immagine di supporto è una copia completa di un oggetto IBM MQ registrato nel giornale. Alcuni oggetti corrotti o danneggiati possono essere automaticamente recuperati dalla loro immagine multimediale.

Un oggetto IBM MQ di lunga durata può rappresentare un gran numero di voci di giornale, tornando al punto in cui è stato creato. Per evitare ciò, IBM MQ for IBM i ha il concetto di immagine multimediale di un oggetto.

Questa immagine di supporto è una copia completa dell'oggetto IBM MQ registrato nel giornale. Se viene presa un'immagine di un oggetto, l'oggetto può essere ricostruito riproducendo le voci di giornale da questa immagine in poi. La voce nel giornale che rappresenta il punto di ripetizione per ogni oggetto IBM MQ viene indicata come voce di recupero del supporto. IBM MQ tiene traccia di:

- Voce di ripristino supporto per ciascun oggetto gestore code.
- La voce meno recente da questa serie (per i dettagli, consultare il messaggio di errore AMQ7462 in [“Gestione giornale su IBM i”](#) a pagina 375).

Le immagini dell'oggetto \*CTLG e dell'oggetto \*MQM vengono prese regolarmente perché questi oggetti sono fondamentali per il riavvio del gestore code.

Le immagini di altri oggetti vengono prese quando è conveniente. Per impostazione predefinita, le immagini di tutti gli oggetti vengono prese quando un gestore code viene arrestato utilizzando il comando **ENDMQM** con il parametro ENDCCTJOB (\*YES). Questa operazione può richiedere una quantità di tempo considerevole per i gestori code molto grandi. Se è necessario chiudere rapidamente, specificare il parametro RCDMQMIMG (\*NO) con ENDCCTJOB (\*YES). In questi casi, si consiglia di registrare

un'immagine di supporto completa nei journal dopo il riavvio del gestore code, utilizzando il seguente comando:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ registra automaticamente un'immagine di un oggetto, se trova un punto conveniente in cui un oggetto può essere descritto in modo compatto da una piccola voce nel giornale. Tuttavia, ciò potrebbe non verificarsi mai per alcuni oggetti, ad esempio, code che contengono in modo congruente un numero elevato di messaggi.

Invece di consentire alla data della voce di ripristino del supporto meno recente di continuare per un periodo inutilmente lungo, utilizzare il comando IBM MQ RCDMQMIMG, che consente di eseguire manualmente un'immagine degli oggetti selezionati.

## Recupero da immagini multimediali

IBM MQ recupera automaticamente alcuni oggetti dalla relativa immagine del supporto se viene rilevato che sono danneggiati o danneggiati. In particolare, ciò si applica agli speciali oggetti \*MQM e \*CTLG come parte del normale avvio del gestore code. Se una transazione del punto di sincronizzazione era incompleta al momento dell'ultimo arresto del gestore code, anche qualsiasi coda interessata viene ripristinata automaticamente, al fine di completare l'operazione di avvio.

È necessario ripristinare altri oggetti manualmente, utilizzando il IBM MQ comando RCRMQMOBJ. Questo comando riproduce le voci nel giornale per ricreare l'oggetto IBM MQ. Se un oggetto IBM MQ viene danneggiato, le uniche azioni valide sono eliminarlo o ricrearlo con questo metodo. Notare, tuttavia, che i messaggi non persistenti non possono essere recuperati in questo modo.

### **Checkpoint su IBM MQ for IBM i**

I punti di controllo vengono utilizzati in vari momenti per fornire un punto di partenza congruente noto per il ripristino.

Il thread del checkpoint all'interno del processo AMQZMUC0 è responsabile dell'acquisizione del punto di controllo nei punti seguenti:

- Avvio del gestore code (STRMQM).
- Chiusura del gestore code (ENDMQM).
- Dopo che è trascorso un periodo di tempo dall'ultimo punto di controllo (il periodo predefinito è 30 minuti) e un numero minimo di record di log è stato scritto dal punto di controllo precedente (il valore predefinito è 100).
- Dopo che è stato scritto un numero di record di log. Il valore predefinito è 10 000.
- Dopo che la dimensione della soglia del giornale è stata superata e un nuovo ricevitore di giornale è stato creato automaticamente.
- Quando un'immagine del supporto completo viene presa con:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

### **Backup dei dati IBM MQ for IBM i**

Utilizzare queste informazioni per comprendere i due tipi di backup IBM MQ per ogni gestore code.

Per ogni gestore code, esistono due tipi di backup IBM MQ da considerare:

- Backup di dati e journal.

Per garantire che entrambe le serie di dati siano congruenti, eseguire questa operazione solo dopo aver chiuso il gestore code.

- Backup journal.

È possibile eseguire questa operazione mentre il gestore code è attivo.

Per entrambi i metodi, è necessario trovare i nomi della directory IFS del gestore code e della libreria del gestore code. È possibile trovarle nel file di configurazione IBM MQ (mq.ini). Per ulteriori informazioni, consultare [La stanza QueueManager](#).

Utilizzare le procedure riportate di seguito per eseguire entrambi i tipi di backup:

### Backup di dati e journal di uno specifico gestore code

**Nota: Non utilizzare una richiesta salva - mentre - attivo quando il gestore code è in esecuzione. Tale richiesta non può essere completata a meno che non venga eseguito il commit o il rollback di tutte le definizioni di commit con modifiche in sospeso. Se questo comando viene utilizzato quando il gestore code è attivo, le connessioni del canale potrebbero non terminare normalmente. Utilizzare sempre la seguente procedura.**

1. Creare un ricevitore di giornale vuoto, utilizzando il comando:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Utilizzare il comando **RCDMQMIMG** per registrare un'immagine di MQM per tutti gli oggetti IBM MQ , quindi forzare un punto di controllo utilizzando il comando:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Terminare i canali e verificare che il gestore code non sia in esecuzione. Se il gestore code è in esecuzione, arrestarlo con il comando **ENDMQM** .
4. Eseguire il backup della libreria del gestore code immettendo il seguente comando:

```
SAVLIB LIB(QMTEST)
```

5. Eseguire il backup delle directory IFS del gestore code immettendo il seguente comando:

```
SAV DEV(...) OBJ(('QIBM/UserData/mqm/qmgrs/test'))
```

### Backup journal di un particolare gestore code

Poiché tutte le informazioni rilevanti sono conservate nei giornali, finché si esegue un salvataggio completo in un determinato momento, le copie di riserva parziali possono essere eseguite salvando i ricevitori di giornale. Queste registrano tutte le modifiche dall'ora del backup completo e vengono eseguite immettendo i seguenti comandi:

1. Creare un ricevitore di giornale vuoto, utilizzando il comando:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Utilizzare il comando **RCDMQMIMG** per registrare un'immagine di MQM per tutti gli oggetti IBM MQ , quindi forzare un punto di controllo utilizzando il comando:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Salvare i ricevitori di giornale utilizzando il seguente comando:

```
SAV0BJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

Una semplice strategia di backup consiste nell'eseguire un backup completo delle librerie IBM MQ ogni settimana ed eseguire un backup journal giornaliero. Questo, ovviamente, dipende da come hai impostato la tua strategia di backup per la tua azienda.

Come parte della strategia di backup, prendersi cura dei ricevitori di giornale. È utile eliminare i ricevitori di giornale dalle librerie IBM MQ per vari motivi:

- Per liberare spazio; questo si applica a tutti i ricevitori di giornale
- Per migliorare le prestazioni all'avvio (STRMQM)
- Per migliorare le prestazioni della ricreazione di oggetti (RCRMQMOBJ)

Prima di cancellare un ricevitore di giornale, è necessario fare attenzione che si disponga di una copia di backup e che non sia più necessario il ricevitore di giornale.

I ricevitori di journal possono essere rimossi dalla libreria del gestore code *dopo* che sono stati scollegati dai journal e salvati, purché siano disponibili per il ripristino se necessario per un'operazione di recupero.

Il concetto di gestione del giornale viene mostrato in [Figura 26 a pagina 376](#).

WebSphere MQ  
for iSeries  
Journal

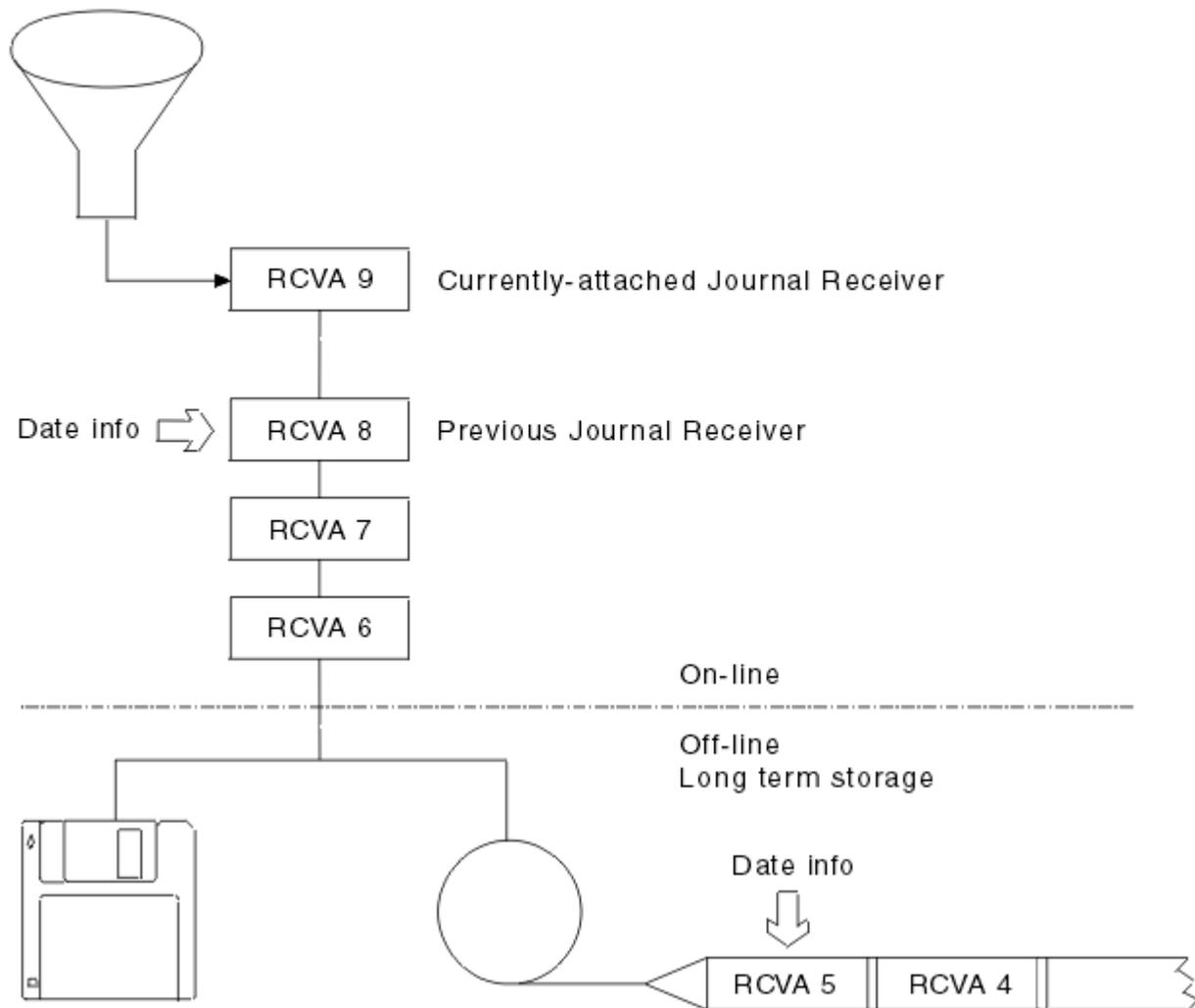


Figura 26. Registrazione su giornale su IBM i

È importante sapere fino a che punto nei journal IBM MQ è probabile che sia necessario, al fine di determinare quando un ricevitore di giornale di cui è stato eseguito il backup può essere rimosso dalla libreria del gestore code e quando il backup stesso può essere eliminato.

IBM MQ invia due messaggi alla coda messaggi del gestore code (QMOMMSG nella libreria del gestore code) per determinare questa ora. Questi messaggi vengono emessi all'avvio, quando si modifica un

ricevitore di giornale locale e si utilizza RCDMQIMG per forzare un punto di controllo. I due messaggi sono:

#### **AMQ7460**

Punto di ripristino avvio. Questo messaggio definisce la data e l'ora della voce di avvio da cui IBM MQ riproduce il giornale nel caso di un passaggio di ripristino di avvio. Se il ricevitore di giornale che contiene questo record è disponibile nelle librerie IBM MQ , questo messaggio contiene anche il nome del ricevitore di giornale che contiene il record.

#### **AMQ7462**

Voce di ripristino del supporto meno recente. Questo messaggio definisce la data e l'ora della voce meno recente da utilizzare per ricreare un oggetto dalla relativa immagine del supporto.

Il ricevitore di giornale identificato è quello più vecchio richiesto. Tutti gli altri ricevitori di giornale IBM MQ con date di creazione precedenti non sono più necessari. Se vengono visualizzate solo le stelle, è necessario ripristinare i backup dalla data indicata per determinare quale è il ricevitore di giornale più vecchio.

Quando questi messaggi vengono registrati, IBM MQ scrive anche un oggetto spazio utente nella libreria del gestore code che contiene solo una voce: il nome del ricevitore di giornale più vecchio che deve essere conservato nel sistema. Questo spazio utente è denominato AMQJRNINF e i dati vengono scritti nel formato:

```
JJJJJJJJJLLLLLLLLLLLLYYYYMMDDHHMSSmmm
```

dove:

#### **JJJJJJJJJ**

È il nome del destinatario meno recente di cui IBM MQ ha ancora bisogno.

#### **LLLLLLLLL**

Indica il nome della libreria del ricevitore di giornale.

#### **YYYY**

Indica l'anno della voce di journal più vecchia necessaria a IBM MQ .

#### **MM**

È il mese della voce di journal più vecchia di cui IBM MQ ha bisogno.

#### **DD**

Indica il giorno della voce di journal più vecchia di cui IBM MQ ha bisogno.

#### **HH**

Indica l'ora della voce di journal meno recente necessaria a IBM MQ .

#### **SS**

Indica i secondi della voce di journal più vecchia di cui IBM MQ ha bisogno.

#### **mmm**

Indica i millisecondi della voce di journal più vecchia di cui IBM MQ ha bisogno.

Quando il ricevitore di giornale più vecchio è stato cancellato dal sistema, questo spazio utente contiene asterischi (\*) per il nome del ricevitore di giornale.

**Nota:** L'esecuzione periodica di RCDMQIMG OBJ(\*ALL) OBJTYPE(\*ALL) DSPJRNDTA(\*YES) consente di risparmiare tempo di avvio per IBM MQ e di ridurre il numero di ricevitori di giornale locali necessari per salvare e ripristinare per il ripristino.

IBM MQ for IBM i non fa riferimento ai ricevitori di giornale a meno che non stia eseguendo un passaggio di ripristino per l'avvio o per la ricreazione di un oggetto. Se rileva che un journal richiesto non è presente, emette il messaggio AMQ7432 alla coda messaggi del gestore code (QMOMMSG), riportando l'ora e la data della voce di journal richiesta per completare il passaggio di recupero.

In questo caso, ripristinare tutti i ricevitori di giornale che sono stati scollegati dopo questa data dal backup, per consentire la riuscita del passaggio di ripristino.

Mantenere il ricevitore di giornale che contiene la voce di avvio e i successivi ricevitori di giornale disponibili nella libreria del gestore code.

Mantenere il ricevitore di giornale contenente il più vecchio Media Recovery Entrye tutti i successivi ricevitori di giornale, sempre disponibili e presenti nella libreria del gestore code o di cui è stato eseguito il backup.

Quando si forza un punto di controllo:

- Se il ricevitore di giornale denominato in AMQ7460 non è avanzato, ciò indica che è necessario eseguire il commit o il rollback di un'unità di lavoro incompleta.
- Se il ricevitore di giornale denominato in AMQ7462 non è avanzato, ciò indica che sono presenti uno o più oggetti danneggiati.

## **Ripristino di un gestore code completo (dati e journal) su IBM i**

Utilizzare queste informazioni per ripristinare uno o più gestori code da un backup o da una macchina remota.

Se è necessario ripristinare uno o più gestori code IBM MQ da un backup, attenersi alla seguente procedura.

1. Disattivare i gestori code IBM MQ .
2. Individuare la serie di backup più recente, costituita dal backup completo più recente e, successivamente, dai ricevitori di giornale.
3. Eseguire un'operazione RSTLIB, dal backup completo, per ripristinare le librerie di dati IBM MQ al loro stato al momento del backup completo, immettendo i seguenti comandi:

```
RSTLIB LIB(QMQRLIB1) .....  
RSTLIB LIB(QMQRLIB2) .....
```

Se un ricevitore di giornale è stato parzialmente salvato in un backup di giornale e completamente salvato in un backup successivo, ripristinare solo quello completamente salvato. Ripristinare i journal singolarmente, in ordine cronologico.

4. Eseguire un'operazione RST per ripristinare gli indirizzari IFS IBM MQ sul file system IFS, utilizzando il seguente comando:

```
RST DEV(...) OBJ((' /QIBM/UserData/mqm/qmgrs/testqm')) ...
```

5. Avviare il gestore code messaggi. Ciò riproduce tutti i record del journal scritti a partire dal backup completo e ripristina tutti gli oggetti IBM MQ allo stato congruente al momento del backup del journal.

Se si desidera ripristinare un gestore code completo su una macchina diversa, utilizzare la seguente procedura per ripristinare tutto dalla libreria del gestore code. (Utilizziamo TEST come nome del gestore code di esempio.)

1. CRTMQM TEST
2. DLTLIB LIB(QMTEST)
3. RSTLIB SAVLIB(QMTEST) DEV(\*SAVF) SAVF(QMGRLIBSAV)
4. Eliminare i seguenti file IFS:

```
/QIBM/UserData/mqm/qmgrs/TEST/QMQMCHKPT  
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMQMOBJCAT  
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMANAGER  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.AUTH.DATA.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CHANNEL.INITQ/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.COMMAND.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.REPOSITORY.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.TRANSMIT.QUEUE/q
```

```
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.PENDING.DATA.QUEUE/q
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.ADMIN.COMMAND.QUEUE/q
```

5. STRMQM TEST

6. RCRMQMOBJ OBJ(\*ALL) OBJTYPE(\*ALL) MQMNAME(TEST)

## **Ripristino dei ricevitori di giornale per un determinato gestore code su IBM i**

Utilizzare queste informazioni per comprendere i diversi modi per ripristinare i ricevitori di giornale.

L'azione più comune consiste nel ripristinare un ricevitore di giornale di cui è stato eseguito il backup in una libreria del gestore code, se un ricevitore che è stato rimosso è necessario di nuovo per una funzione di ripristino successiva.

Si tratta di un'attività semplice e richiede il ripristino dei ricevitori di giornale utilizzando il comando RSTOBJ IBM i standard:

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNRVC) .....
```

Potrebbe essere necessario ripristinare una serie di ricevitori di giornale, piuttosto che un singolo ricevitore. Ad esempio AMQA000007 è il ricevitore più vecchio nelle librerie IBM MQ e sia AMQA000005 che AMQA000006 devono essere ripristinati.

In questo caso, ripristinare i ricevitori singolarmente in ordine cronologico inverso. Non è sempre necessario, ma è una buona pratica. In situazioni gravi, potrebbe essere necessario utilizzare il IBM i comando WRKJRNA per associare i ricevitori di giornale ripristinati al giornale.

Quando si ripristinano i giornali, il sistema crea automaticamente un ricevitore di giornale collegato con un nuovo nome nella sequenza del ricevitore di giornale. Tuttavia, il nuovo nome creato potrebbe essere lo stesso di un ricevitore di giornale che è necessario ripristinare. L'intervento manuale è necessario per risolvere questo problema; per creare un nuovo ricevitore di giornale in sequenza e un nuovo giornale prima di ripristinare il ricevitore di giornale.

Ad esempio, considerare il problema con il giornale salvato AMQAJRN e i seguenti ricevitori di giornale:

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

Quando si ripristina il journal AMQAJRN in una libreria del gestore code, il sistema crea automaticamente il ricevitore di journal AMQA000000. Questo ricevitore generato automaticamente è in conflitto con uno dei ricevitori di giornale esistenti (AMQA000000) che si desidera ripristinare e che non è possibile ripristinare.

La soluzione è:

1. Creare manualmente il ricevitore di giornale successivo (consultare [“Journal del gestore code su IBM i” a pagina 370](#)):

```
CRTJRNRVC JRNRVC(QMQRLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Creare manualmente il giornale con il ricevitore di giornale:

```
CRTJRN JRN(QMGRLIB/AMQAJRN) MNGRCV(*SYSTEM) +
JRNRVC(QMGRLIB/AMQA9000001) MSGQ(QMGRLIB/AMQAJRNMSG)
```

3. Ripristinare i ricevitori di giornale locali AMQA000000 in AMQA900000.

## IBM i Gestori code a più istanze su IBM i

I gestori code a più istanze migliorano la disponibilità passando automaticamente a un server standby se il server attivo ha esito negativo. I server attivo e standby sono più istanze dello stesso gestore code; condividono gli stessi dati del gestore code. Se l'istanza attiva non riesce, è necessario trasferire il proprio journal allo standby che assume il controllo in modo che il gestore code possa ricreare le proprie code.

Configurare i sistemi IBM i su cui sono in esecuzione gestori code a più istanze in modo che, se l'istanza del gestore code attiva ha esito negativo, il journal che sta utilizzando sia disponibile per l'istanza in standby che assume il controllo. È possibile progettare le proprie attività di configurazione e amministrazione per rendere il journal dall'istanza attiva disponibile per l'istanza che assume il controllo. Se non si desidera perdere i messaggi, la progettazione deve garantire che il giornale di standby sia congruente con il giornale attivo nel punto di errore. È possibile adattare la propria progettazione da una delle due configurazioni descritte con esempi negli argomenti successivi che mantengono la coerenza.

1. Eseguire il mirroring del journal dal sistema su cui è in esecuzione l'istanza del gestore code attivo ai sistemi su cui sono in esecuzione istanze in standby.
2. Collocare il giornale in un IASP (Independent Auxiliary Storage Pool) trasferibile dal sistema che esegue l'istanza attiva ad un'istanza in standby.

La prima soluzione non richiede hardware o software aggiuntivi poiché utilizza gli ASP di base. La seconda soluzione richiede IASP commutabili che necessitano del supporto cluster IBM i disponibile come IBM i License Product 5761-SS1 Opzione 41 a parte.

## IBM i Affidabilità e disponibilità su IBM i

I gestori code a più istanze mirano a migliorare la disponibilità delle applicazioni. I vincoli fisici e tecnologici richiedono soluzioni diverse per rispondere alle esigenze di disaster recovery, backup dei gestori code e operazioni continue.

Nella configurazione per l'affidabilità e la disponibilità si scambiano un gran numero di fattori, con il risultato di quattro punti di progettazione distinti:

### Ripristino di emergenza

Ottimizzato per il ripristino dopo un grave disastro che distrugge tutti gli asset locali.

Il ripristino di emergenza su IBM i è spesso basato sul mirroring geografico dell'IASP.

### Backup

Ottimizzato per il recupero dopo un guasto localizzato, di solito un errore umano o qualche problema tecnico imprevisto.

IBM MQ fornisce gestori code di backup per eseguire periodicamente il backup dei gestori code. È anche possibile utilizzare la replica asincrona dei journal del gestore code per migliorare la valuta del backup.

### Disponibilità

Ottimizzato per ripristinare rapidamente le operazioni, offrendo l'aspetto di un servizio quasi ininterrotto a seguito di prevedibili errori tecnici come un server o un errore del disco.

Il recupero è in genere misurato in minuti, con il rilevamento che a volte richiede più tempo del processo di ripristino. Un gestore code a più istanze ti assiste nella configurazione per la *disponibilità*.

### Funzionamento continuo

Ottimizzato per fornire un servizio ininterrotto.

Le soluzioni di funzionamento continuo devono risolvere il problema di rilevamento, e quasi sempre comportano la presentazione dello stesso lavoro attraverso più di un sistema e utilizzando il primo risultato, o se la correttezza è una considerazione importante, confrontando almeno due risultati.

Un gestore code a più istanze ti assiste nella configurazione per la *disponibilità*. Un'istanza del gestore code è attiva alla volta. Il passaggio a un'istanza standby richiede da poco più di dieci secondi a quindici minuti o più, a seconda di come il sistema è configurato, caricato e regolato.

Un gestore code a più istanze può fornire l'aspetto di un servizio quasi ininterrotto se utilizzato con IBM MQ MQI clients ricollegabile, che è in grado di continuare l'elaborazione senza che il programma applicativo sia necessariamente consapevole di un'interruzione del gestore code; consultare l'argomento [Riconnessione client automatizzata](#).

IBM i

## **Componenti di una soluzione alta disponibilità su IBM i**

Creare una soluzione di alta disponibilità utilizzando i gestori code a più istanze fornendo una memoria di rete solida per i dati del gestore code, la replica del journal o la memoria IASP robusta per i journal del gestore code e utilizzando i client ricollegabili, delle applicazioni configurate come servizi del gestore code riavviabili.

Un gestore code a più istanze reagisce al rilevamento dell'errore del gestore code riavviando un'altra istanza del gestore code su un altro server. Per completare l'avvio, l'istanza deve accedere ai dati del gestore code condiviso nella memoria di rete e alla relativa copia del journal del gestore code locale.

Per creare una soluzione alta disponibilità, è necessario gestire la disponibilità dei dati del gestore code, la valuta del journal del gestore code locale e creare applicazioni client riconnettibili oppure distribuire le proprie applicazioni come servizi del gestore code per riavviare automaticamente quando il gestore code viene ripristinato. La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

### **Dati del gestore code**

Posizionare i dati del gestore code su una memoria di rete condivisa, altamente disponibile e affidabile, possibilmente utilizzando dischi RAID di livello 1 o superiore. Il file system deve soddisfare i requisiti per un file system condiviso per i gestori code a più istanze; per ulteriori informazioni sui requisiti per i file system condivisi, consultare [Requisiti per i file system condivisi](#). Network File System 4 (NFS4) è un protocollo che soddisfa questi requisiti.

### **Journal del gestore code**

È inoltre necessario configurare i journal IBM i utilizzati dalle istanze del gestore code in modo che l'istanza in standby sia in grado di ripristinare i dati del gestore code in uno stato congruente. Per un servizio ininterrotto, ciò significa che è necessario ripristinare i journal al loro stato quando l'istanza attiva ha avuto esito negativo. A differenza delle soluzioni di backup o di ripristino di emergenza, il ripristino dei journal in un punto di controllo precedente non è sufficiente.

Non è possibile condividere fisicamente i journal tra più sistemi IBM i sulla memoria di rete. Per ripristinare i journal del gestore code allo stato congruente nel punto in cui si è verificato l'errore, è necessario trasferire il journal fisico locale all'istanza del gestore code attivo al momento dell'errore alla nuova istanza che è stata attivata oppure un mirror di gestione del journal sulle istanze in standby in esecuzione. Il giornale sottoposto a mirroring è una replica del giornale remoto che è stata mantenuta esattamente sincronizzata con il giornale locale appartenente all'istanza non riuscita.

Tre configurazioni sono i punti di partenza per la progettazione della modalità di gestione dei journal per un gestore code a più istanze,

1. Utilizzo della replica del giornale sincronizzata (mirroring del giornale) dall'ASP dell'istanza attiva agli ASP delle istanze in standby.
2. Trasferimento di un IASP configurato per conservare il journal del gestore code dall'istanza attiva all'istanza in standby che sta subendo il controllo come istanza attiva.
3. Utilizzo dei mirror IASP secondari sincronizzati.

Consultare le opzioni [ASP](#), per ulteriori informazioni sull'inserimento dei dati del gestore code in un IASP, nel comando CRTMQM di IBM MQ IBM i.

Consultare inoltre [Alta disponibilità](#) nelle IBM i informazioni in IBM Documentation.

## Applicazioni

Per creare un client per la riconnessione automatica al gestore code quando il gestore code in standby riprende la normale attività, connettere l'applicazione al gestore code utilizzando MQCONNX e specificare MQCNO\_RECONNECT\_Q\_MGR nel campo **MQCNO** Opzioni . Consultare [Programmi di esempio di alta disponibilità per tre programmi di esempio che utilizzano client ricollegabili](#) e [Ripristino applicazione per informazioni sulla progettazione di applicazioni client per il ripristino](#).

 *Creazione di una condivisione di rete per i dati del gestore code utilizzando NetServer su IBM i*  
Creare una condivisione di rete su un server IBM i per memorizzare i dati del gestore code. Configurare le connessioni da due server, che ospiteranno le istanze del gestore code, per accedere alla condivisione di rete.

## Prima di iniziare

- Per questa attività sono richiesti tre server IBM i . La condivisione di rete è definita su uno dei server, GAMMA. Gli altri due server, ALPHA e BETA, devono connettersi a GAMMA.
- Installare IBM MQ su tutti e tre i server.
- Installare System i Navigator; consultare [System i Navigator](#).

## Informazioni su questa attività

- Creare la directory del gestore code su GAMMA e impostare la proprietà e le autorizzazioni corrette per i profili utente QMQM e QMQMADM. La directory e le autorizzazioni sono facilmente create installando IBM MQ su GAMMA.
- Utilizzare System i Navigator per creare una condivisione nella directory dei dati del gestore code su GAMMA.
- Creare directory su ALPHA e BETA che puntino alla condivisione.

## Procedura

1. In GAMMA, creare la directory per ospitare i dati del gestore code con il profilo utente QMQM come proprietario e QMQMADM come gruppo principale.

### Suggerimento:

Un modo rapido e affidabile per creare la directory con i permessi corretti è installare IBM MQ su GAMMA.

Successivamente, se non si desidera eseguire IBM MQ su GAMMA, disinstallare IBM MQ. Dopo la disinstallazione, la directory /QIBM/UserData/mqm/qmgrs rimane su GAMMA con il profilo utente QMQM del proprietario e QMQMADM il gruppo primario.

L'attività utilizza la directory /QIBM/UserData/mqm/qmgrs su GAMMA per la condivisione.

2. Avviare la procedura guidata System i Navigator **Aggiungi connessione** e connettersi al sistema GAMMA.
  - a) Fare doppio clic sull'icona **System i Navigator** sul desktop Windows .
  - b) Fare clic su **Si** per creare una connessione.
  - c) Seguire le istruzioni nella procedura guidata **Aggiungi connessione** e creare una connessione dal sistema IBM i a GAMMA.

La connessione a GAMMA viene aggiunta a **Connessioni**.

3. Aggiungere una nuova condivisione file su GAMMA.
  - a) Nella finestra **System i Navigator** , fare clic sulla cartella File Shares in My Connections/ GAMMA/File Systems.
  - b) Nella finestra **Attività personali** , fare clic su **Gestisci condivisioni IBM i NetServer**.

Una nuova finestra, **IBM i NetServer - GAMMA**, si apre sul desktop e mostra oggetti condivisi.

- c) Fare clic con il tasto destro del mouse sulla cartella Shared Objects > **file** > **Nuovo** > **File**.  
Viene visualizzata una nuova finestra, **IBM i NetServer File Share - GAMMA**.
- d) Fornire un nome alla condivisione, ad esempio WMQ .
- e) Impostare il controllo accessi su Read/Write.
- f) Selezionare il **Nome percorso** passando alla directory /QIBM/UserData/mqm/qmgrs creata in precedenza e fare clic su **OK**.  
La finestra **IBM i NetServer Condivisione file - GAMMA** viene chiusa e WMQ viene elencato nella finestra degli oggetti condivisi.
4. Fare clic con il pulsante destro del mouse su **WMQ** nella finestra degli oggetti condivisi. Fare clic su **File** > **Autorizzazioni**.  
Si apre una finestra, **Qmgrs Permissions - GAMMA**, per l'oggetto /QIBM/UserData/mqm/qmgrs.
- a) Controllare le seguenti autorizzazioni per QMQM, se non sono già impostate:
- Read
  - Write
  - Execute
  - Management
  - Existence
  - Alter
  - Reference
- b) Controllare le seguenti autorizzazioni per QMQMADM, se non sono già impostate:
- Read
  - Write
  - Execute
  - Reference
- c) Aggiungere altri profili utente a cui si desidera concedere le autorizzazioni a /QIBM/UserData/mqm/qmgrs.  
Ad esempio, è possibile fornire le autorizzazioni predefinite del profilo utente (Pubblico) Read e Execute a /QIBM/UserData/mqm/qmgrs.
5. Verificare che tutti i profili utente a cui è stato concesso l'accesso a /QIBM/UserData/mqm/qmgrs su GAMMA abbiano la stessa parola d'ordine dei server che accedono a GAMMA.  
In particolare, assicurarsi che i profili utente QMQM su altri server, che accederanno alla condivisione, abbiano la stessa password del profilo utente QMQM su GAMMA.
- Suggerimento:** Fare clic sulla cartella My Connections/GAMMA/Users and Groups in System i Navigator per impostare la password. In alternativa, utilizzare i comandi **CHFUSRPRF** e **CHGPWD** .

## Risultati

Verificare di poter accedere a GAMMA da altri server utilizzando la condivisione. Se si stanno eseguendo altre attività, verificare di poter accedere a GAMMA da ALPHA e BETA utilizzando il percorso /QNTC/GAMMA/WMQ. Se la directory /QNTC/GAMMA non esiste in ALPHA o BETA, è necessario creare la directory. A seconda del dominio NetServer , potrebbe essere necessario eseguire l'IPL ALPHA o BETA prima di creare la directory.

```
CRTDIR DIR('/QNTC/GAMMA')
```

Una volta verificato che si ha accesso a /QNTC/GAMMA/WMQ da ALPHA o BETA, immettendo il comando, CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ') crea /QIBM/UserData/mqm/qmgrs/QM1 su GAMMA.

## Operazioni successive

Creare un gestore code a più istanze effettuando le operazioni riportate nelle attività [“Creazione di un gestore code a più istanze utilizzando il mirroring del journal e NetServer su IBM i” a pagina 395](#) o [“Conversione di un gestore code a istanza singola in un gestore code a più istanze utilizzando NetServer e il mirroring del journal su IBM i” a pagina 399](#).

### **Prestazioni di failover su IBM i**

Il tempo impiegato per rilevare un'istanza del gestore code ha avuto esito negativo e quindi per riprendere l'elaborazione su uno standby può variare da decine di secondi a quindici minuti o più a seconda della configurazione. Le prestazioni devono essere una considerazione importante nella progettazione e nel test di una soluzione alta disponibilità.

Vi sono vantaggi e svantaggi da valutare quando si decide se configurare un gestore code a più istanze per utilizzare la replica del journal o per utilizzare un IASP. Il mirroring richiede che il gestore code scriva in modo sincrono su un journal remoto. Da un punto di vista hardware, non è necessario che ciò influisca sulle prestazioni, ma da un punto di vista software, la scrittura su un giornale remoto comporta una lunghezza del percorso maggiore rispetto a quella su un giornale locale e ciò potrebbe ridurre in qualche misura le prestazioni di un gestore code in esecuzione. Tuttavia, quando il gestore code in standby assume il controllo, il ritardo nella sincronizzazione del relativo journal locale dal journal remoto gestito dall'istanza attiva prima che abbia esito negativo, è generalmente piccolo rispetto al tempo impiegato da IBM i per rilevare e trasferire l'IASP sul server che esegue l'istanza in standby del gestore code. I tempi di trasferimento IASP possono essere da dieci a quindici minuti invece di essere completati in secondi. Il tempo di trasferimento IASP dipende dal numero di oggetti che devono essere *attivati* quando l'IASP viene trasferito al sistema di standby e dalla dimensione dei percorsi di accesso o degli indici che devono essere uniti.

Quando il gestore code in standby assume il controllo, il ritardo nella sincronizzazione del relativo journal locale dal journal remoto gestito dall'istanza attiva prima che abbia esito negativo, è generalmente piccolo rispetto al tempo impiegato da IBM i per rilevare e trasferire l'ASP indipendente sul server che esegue l'istanza in standby del gestore code. I tempi di trasferimento ASP indipendente possono essere da dieci a quindici minuti invece di essere completati in secondi. Il tempo di trasferimento ASP indipendente dipende dal numero di oggetti che devono essere *attivati* quando l'ASP indipendente viene trasferito al sistema standby e dalla dimensione dei percorsi di accesso o degli indici che devono essere uniti.

Tuttavia, il trasferimento del giornale non è l'unico fattore che influenza il tempo impiegato per la ripresa completa dell'istanza in standby. È inoltre necessario considerare il tempo impiegato dal file system di rete per rilasciare il blocco sui dati del gestore code che segnala all'istanza in standby di continuare con l'avvio e anche il tempo impiegato per ripristinare le code dal journal, in modo che l'istanza sia in grado di avviare nuovamente l'elaborazione dei messaggi. Queste altre fonti di ritardo si aggiungono tutte al tempo necessario per avviare un'istanza in standby. Il tempo totale di commutazione è costituito dai seguenti componenti:

#### **Tempo di rilevamento degli errori**

Il tempo impiegato da NFS per rilasciare il blocco sui dati del gestore code e l'istanza in standby per continuare il processo di avvio.

#### **Tempo di trasferimento**

Nel caso di un cluster HA, il tempo impiegato da IBM i per il trasferimento dell'IASP dal sistema che ospita l'istanza attiva all'istanza in standby e, in caso di replica del giornale, il tempo impiegato per aggiornare il giornale locale nello standby con i dati dalla replica remota.

#### **Data/ora di riavvio**

Il tempo impiegato dall'istanza del gestore code appena attiva per ricreare le code dall'ultimo punto di controllo nel journal ripristinato e per riprendere l'elaborazione dei messaggi.

#### **Nota:**

Se l'istanza in standby che ha assunto il controllo è configurata per la replica sincrona sull'istanza precedentemente attiva, l'avvio potrebbe essere ritardato. La nuova istanza attivata potrebbe non essere in grado di eseguire la replica sul giornale remoto, se il giornale remoto si trova sul server che ospitava l'istanza precedentemente attiva e il server ha riportato un errore.

Il tempo di attesa predefinito per una risposta sincrona è un minuto. È possibile configurare il ritardo massimo prima del timeout della replica. In alternativa, è possibile configurare le istanze in standby per l'avvio utilizzando la replica asincrona per l'istanza attiva non riuscita. Successivamente si passa alla replica sincrona, quando l'istanza in errore è nuovamente in esecuzione in standby. La stessa considerazione si applica all'utilizzo di mirroring ASP indipendenti sincroni.

È possibile effettuare misurazioni di baseline separate per questi componenti per consentire di valutare il tempo complessivo per il failover e per determinare quale approccio di configurazione utilizzare. Nel prendere la migliore decisione di configurazione, è anche necessario considerare il modo in cui altre applicazioni sullo stesso server eseguiranno il failover e se vi sono processi di backup o di ripristino di emergenza che utilizzano già l'IASP.

I tempi di trasferimento IASP possono essere ridotti ottimizzando la configurazione del cluster:

1. I profili utente tra i sistemi nel cluster devono avere lo stesso GID e UID per eliminare la necessità per il processo di attivazione di modificare UID e GID.
2. Ridurre al minimo il numero di oggetti di database nel sistema e nei lotti dischi utente di base, poiché questi devono essere uniti per creare la tabella di riferimenti incrociati per il gruppo di lotti dischi.
3. Ulteriori suggerimenti sulle prestazioni sono disponibili in IBM Redbook, *Implementazione PowerHA per IBM i*, SG24-7405.

Una configurazione che utilizza gli ASP di base, il mirroring del giornale e una piccola configurazione deve essere commutata nell'ordine di decine di secondi.

## **Panoramica sulla combinazione delle funzionalità di clustering IBM i con il clustering IBM MQ**

L'esecuzione di IBM MQ su IBM i e l'utilizzo delle funzioni di clustering di IBM i possono fornire una soluzione di alta disponibilità più completa, rispetto all'utilizzo del solo clustering IBM MQ.

Per disporre di questa funzione, è necessario impostare:

1. Cluster sulla tua macchina IBM i; vedi [“IBM iCluster” a pagina 385](#)
2. Un IASP (independent auxiliary storage pool), in cui si sposta il gestore code; consultare [“IASP \(Independent Auxiliary Storage Pools\)” a pagina 385](#)
3. Un CRG (cluster resource group); consultare [“Gruppi di risorse cluster unità” a pagina 386](#), in cui si definisce:
  - Dominio di ripristino
  - IASP
  - Programma di uscita; consultare [“Programma di uscita CRG unità” a pagina 386](#)

### **IBM iCluster**

Un cluster IBM i è una raccolta di istanze, ossia computer o partizioni IBM i, che sono collegati logicamente tra loro.

Lo scopo di questo raggruppamento è consentire il backup di ogni istanza, eliminando un singolo punto di errore e aumentando la resilienza dei dati e delle applicazioni. Con un cluster creato, è possibile configurare i vari tipi di CRG (cluster resource group) per gestire applicazioni, dati e unità nel cluster.

Per ulteriori informazioni, consultare [Creazione di un cluster](#) e il comando CRTCLU ([Creazione cluster](#)).

### **IASP (Independent Auxiliary Storage Pools)**

Un IASP è un tipo di ASP utente che serve come estensione della memoria a livello singolo. Si tratta di un pezzo di memoria che, a causa della sua indipendenza dalla memoria di sistema, può essere facilmente manipolato senza dover eseguire l'IPL del sistema.

Un IASP può essere facilmente commutato in un'altra istanza del sistema operativo o replicato in un IASP di destinazione su un'altra istanza del sistema operativo. È possibile utilizzare due metodi per commutare un IASP tra le istanze:

- Il primo metodo richiede che tutti i computer nel cluster e la tower disco commutabile contenente l'IASP, siano collegati utilizzando un loop HSL (High Speed Link).
- Il secondo metodo richiede che le istanze del sistema operativo siano partizioni sullo stesso computer IBM i in cui gli IOP (input/output processor) possono essere commutati tra le partizioni. Non è necessario alcun hardware speciale per poter replicare un IASP. La replica viene eseguita utilizzando TCP/IP sulla rete.

Per ulteriori informazioni, consultare il comando [Configurazione ASP unità \(CFGDEVASP\)](#).

## Gruppi di risorse cluster unità

Esistono diversi tipi di CRG (cluster resource group). Per ulteriori informazioni sui diversi tipi di CRG disponibili, consultare [Gruppo di risorse cluster](#).

Questo argomento si concentra su un CRG unità. Un CRG unità:

- Descrive e gestisce le risorse delle unità come gli IASP (independent auxiliary storage pools).
- Definisce il dominio di ripristino dei nodi cluster
- Assegna un dispositivo e
- Assegna il programma di uscita che gestirà gli eventi cluster.

Il dominio di ripristino indica quale nodo cluster verrà considerato come nodo primario. Il resto dei nodi sono considerati backup. I nodi di backup vengono ordinati anche nel dominio di ripristino, specificando quale nodo è il primo backup, il secondo backup e così via, a seconda del numero di nodi presenti nel dominio di ripristino.

In caso di errore del nodo primario, il programma di uscita viene eseguito su tutti i nodi nel dominio di ripristino. Il programma di uscita in esecuzione sul primo backup può quindi eseguire le inizializzazioni necessarie per rendere questo nodo il nuovo nodo primario.

Consultare [Creazione CRG unità](#) e il comando [Creazione CRG \(Cluster Resource Group\)](#) per ulteriori informazioni.

## Programma di uscita CRG unità

Il servizio di risorse cluster del sistema operativo richiama un programma di uscita CRG unità quando si verifica un evento in uno dei nodi definiti dal dominio di ripristino; ad esempio, un evento di failover o di commutazione.

Un evento di failover si verifica quando il nodo primario del cluster ha esito negativo e i CRG vengono commutati con tutte le risorse che gestiscono, mentre un evento di commutazione si verifica quando un CRG specifico viene commutato manualmente dal nodo primario al nodo di backup.

In entrambi i casi, il programma di uscita è responsabile dell'inizializzazione e dell'avvio di tutti i programmi in esecuzione sul precedente nodo primario, che converte il primo nodo di backup nel nuovo nodo primario.

Ad esempio, con IBM MQ, il programma di uscita deve essere responsabile dell'avvio del sottosistema IBM MQ (QMQM) e dei gestori code. I gestori code devono essere configurati per avviare automaticamente listener e servizi, ad esempio i controlli dei trigger.

Un programma di uscita di esempio, AMQSCR4, è disponibile su IBM i all'indirizzo IBM MQ 9.1

## Configurazione IASP commutabile

IBM MQ può essere impostato per sfruttare le funzionalità di clustering di IBM i. Per far ciò:

1. Creare un cluster IBM i tra i sistemi del data center

2. Spostare il gestore code in un IASP.

“Spostamento o rimozione di un gestore code in o da un ASP (auxiliary storage pool) indipendente” a pagina 388 contiene del codice di esempio che consente di eseguire questa operazione.

3. È necessario creare un CRG definendo il dominio di ripristino, l'IASP e il programma di uscita.

“Configurazione di un CRG (cluster resource group) unità” a pagina 387 contiene del codice di esempio che consente di eseguire questa operazione.

### Concetti correlati

“ASP indipendenti e alta disponibilità” a pagina 408

Gli ASP indipendenti consentono alle applicazioni e ai dati di essere spostati tra i server. La flessibilità degli ASP indipendenti significa che sono la base per alcune soluzioni di alta disponibilità IBM i . Nel considerare se utilizzare un ASP o un ASP indipendente per il journal del gestore code, è necessario considerare un'altra configurazione ad alta disponibilità basata su ASP indipendenti.

 *Configurazione di un CRG (cluster resource group) unità*

Un programma di esempio per impostare un CRG (Cluster Resource Group) unità.

### Informazioni su questa attività

Nel seguente esempio, notare che:

- [ PRIMARY SITE NAME] e [ BACKUP SITE NAME] possono essere due stringhe distinte di otto caratteri o meno.
- [ PRIMARY IP] e [ BACKUP IP] sono gli IP da utilizzare per il mirroring.

### Procedura

1. Identificare il nome del cluster.
2. Identificare il nome del programma di uscita CRG e la libreria.
3. Determinare il nome del nodo primario e i nodi backup che devono essere definiti da questo CRG.
4. Identificare l'IASP che deve essere gestito da questo CRG e assicurarsi che sia stato creato nel nodo primario.
5. Creare una descrizione unità nei nodi di backup utilizzando il seguente comando:

```
CRTDEVASP DEVD([IASP NAME]) RSRNAME([IASP NAME])
```

6. Aggiungere l'indirizzo IP di acquisizione a tutti i nodi utilizzando il seguente comando:

```
ADDTCPICF INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. Avviare l'indirizzo IP di acquisizione solo nel nodo primario utilizzando il comando:

```
STRTCPICF INTNETADR(' [TAKEOVER IP]')
```

8. Opzionale: Se l'IASP è commutabile, richiamare il seguente comando:

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG( [CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT  
NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY) ([BACKUP NAME] *BACKUP))  
EXITPGMFMT(EXTP0200) CFGOBJ(([IAPS NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

9. Opzionale: Se il proprio IASP deve essere sottoposto a mirroring, richiamare questo comando:

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])  
USRPRF([EXIT PROFILE]) RCYDMN(( [PRIMARY NODE] *PRIMARY *LAST [PRIMARY SITE NAME] (' [PRIMARY  
IP] '))  
[BACKUP NAME] *BACKUP *LAST [BACKUP SITE NAME] (' [BACKUP IP] '))) EXITPGMFMT(EXTP0200)  
CFGOBJ(([IAPS NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

Un programma di esempio per spostare un gestore code in un IASP (independent auxiliary storage pool) e comandi per rimuovere un gestore code da uno IASP.

## Informazioni su questa attività

Nel seguente esempio, notare che:

- [MANAGER NAME] è il nome del gestore code.
- [IASP NAME] è il nome dell'IASP.
- [MANAGER LIBRARY] è il nome della libreria del gestore code.
- [MANAGER DIRECTORY] è il nome della tua directory del gestore code.

## Procedura

1. Identificare il nodo primario e i nodi di backup.
2. Eseguire la seguente procedura sul nodo primario:
  - a) Assicurarsi che il gestore code sia terminato.
  - b) Verificare che l'IASP sia vary on utilizzando il comando

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

- c) Creare la directory dei gestori code nell'IASP.  
Ci sarà una directory sotto la root con il nome del tuo IASP, che è:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- d) Spostare gli oggetti IFS del proprio gestore nella directory dei gestori code appena creati nell'IASP utilizzando il seguente comando:

```
QSH CMD('mv /QIBM/UserData/mqm/qmgrs/[MANAGER NAME]
/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- e) Creare un file di salvataggio temporaneo denominato MGRLIB utilizzando il comando:

```
CRTSAVF QGPL/MGRLIB
```

- f) Salvare la libreria del gestore code nel file di salvataggio MGRLIB utilizzando il seguente comando:

```
SAVLIB LIB([MANGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
```

- g) Eliminare la libreria del gestore code utilizzando il seguente comando e ignorare tutti i messaggi di interrogazione:

```
DLTLIB [MANAGER LIBRARY]
```

- h) Ripristinare la libreria del gestore code nell'IASP utilizzando il seguente comando:

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
RSTASPDEV([IASP NAME])
```

- i) Eliminare il file di salvataggio temporaneo utilizzando il seguente comando:

```
DLTF FILE(QGPL/MGRLIB)
```

- j) Creare un collegamento simbolico agli oggetti IFS del gestore code nell'IASP, utilizzando il seguente comando:

```
ADDLNK OBJ('/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER_NAME]')
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER_NAME]')
```

k) Collegarsi all'IASP utilizzando il seguente comando:

```
SETASPGRP [IASP_NAME]
```

l) Avviare il gestore code utilizzando il seguente comando:

```
STRMQM [MANAGER_NAME]
```

3. Eseguire la seguente procedura sul nodo o sui nodi di backup:

a) Creare una directory temporanea del gestore code utilizzando il seguente comando:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER_NAME]')
```

b) Creare un link simbolico alla directory temporanea del gestore code utilizzando il seguente comando:

```
ADDLNK OBJ('/[IASP_NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER_NAME]')
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER_NAME]')
```

c) Eliminare la directory temporanea utilizzando il comando seguente:

```
QSH CMD('rm -r /[IASP_NAME]')
```

d) Aggiungere quanto segue alla fine del file /QIBM/UserData/mqm/mqs.ini:

```
QueueManager:
Name=[MANAGER_NAME]
Prefix=/QIBM/UserData/mqm
Library=[MANAGER_LIBRARY]
Directory=[MANAGER_DIRECTORY]
```

4. Per rimuovere un gestore code da un IASP, immettere i seguenti comandi:

a) VRYCFG CFGOBJ ([ IASP NAME]) CFGTYPE (\*DEV) STATUS (\*ON)

b) SETASPGRP [ NOME IASP]

c) ENDMQM [ NOME GESTORE]

d) DLTMQM [ NOME GESTORE]

## **Configurazione giornale di mirroring per ASP su IBM i**

Configurare un gestore code a più istanze solido utilizzando la replica sincrona tra i journal con mirroring.

Una configurazione del gestore code con mirroring utilizza i journal creati in ASP (auxiliary storage pool) di base o indipendenti.

Su IBM i, i dati del gestore code vengono scritti nei journal e in un filesystem. I journal contengono la copia principale dei dati del gestore code. I giornali sono condivisi tra sistemi che utilizzano la replica del giornale sincrona o asincrona. Per riavviare un'istanza del gestore code è necessaria una combinazione di journal locali e remoti. Il riavvio del gestore code legge i record journal dalla combinazione di journal locali e remoti sul server e i dati del gestore code sul file system di rete condiviso. I dati nel file system accelera il riavvio del gestore code. I punti di controllo vengono memorizzati nel file system, contrassegnando i punti di sincronizzazione tra il file system e i journal. I record del journal memorizzati prima del punto di controllo non sono richiesti per i tipici riavvii del gestore code. Tuttavia, i dati nel file system potrebbero non essere aggiornati e i record del journal dopo il punto di controllo vengono utilizzati per completare il riavvio del gestore code. I dati nei journal collegati all'istanza vengono mantenuti aggiornati in modo che il riavvio possa essere completato correttamente.

Ma anche i record del giornale potrebbero non essere aggiornati, se il giornale remoto sul server standby è stato replicato in modo asincrono e l'errore si è verificato prima della sincronizzazione. Nel caso in cui si

decida di riavviare un gestore code utilizzando un journal remoto non sincronizzato, l'istanza del gestore code in standby potrebbe rielaborare i messaggi che erano stati eliminati prima dell'errore dell'istanza attiva o non elaborare i messaggi che erano stati ricevuti prima dell'errore dell'istanza attiva.

Un'altra, rara possibilità, è che il file system contenga il record del punto di controllo più recente, mentre un giornale remoto non sincronizzato sullo standby non lo fa. In tal caso, il gestore code non viene riavviato automaticamente. È possibile attendere che il journal remoto sia sincronizzato o avviare a freddo il gestore code standby dal file system. Anche se, in questo caso, il file system contiene un punto di controllo più recente dei dati del gestore code rispetto al journal remoto, potrebbe non contenere tutti i messaggi elaborati prima dell'errore dell'istanza attiva. Alcuni messaggi potrebbero essere rielaborati e altri non elaborati, dopo un riavvio a freddo che non è sincronizzato con i giornali.

Con un gestore code a più istanze, il file system viene utilizzato anche per controllare quale istanza di un gestore code è attiva e quale è lo standby. L'istanza attiva acquisisce un blocco sui dati del gestore code. Lo standby attende per acquisire il blocco e quando lo fa, diventa l'istanza attiva. Il blocco viene rilasciato dall'istanza attiva, se termina normalmente. Il blocco viene rilasciato dal sistema di file se il sistema di file rileva che l'istanza attiva ha avuto esito negativo o non può accedere al sistema di file. Il file system deve soddisfare i requisiti per la rilevazione degli errori; consultare [Requisiti dei file system condivisi](#).

L'architettura dei gestori code a più istanze su IBM i fornisce il riavvio automatico dopo un errore del server o del gestore code. Supporta inoltre il ripristino dei dati del gestore code a seguito di un errore del file system in cui sono memorizzati i dati del gestore code.

In [Figura 27 a pagina 391](#), se ALPHA ha esito negativo, è possibile riavviare manualmente QM1 su BETA, utilizzando il giornale sottoposto a mirroring. Aggiungendo la funzionalità del gestore code a più istanze a QM1, l'istanza in standby di QM1 riprende automaticamente su BETA se l'istanza attiva su ALPHA ha esito negativo. QM1 può anche riprendere automaticamente se è il server ALPHA che ha esito negativo, non solo l'istanza attiva di QM1. Una volta che BETA diventa l'host dell'istanza del gestore code attivo, l'istanza in standby può essere avviata su ALPHA.

[Figura 27 a pagina 391](#) mostra una configurazione che esegue il mirroring dei journal tra due istanze di un gestore code utilizzando NetServer per memorizzare i dati del gestore code. È possibile espandere il modello per includere più journal e quindi più istanze. Seguire le regole di denominazione del giornale descritte nell'argomento, [“Journal del gestore code su IBM i” a pagina 370](#). Attualmente il numero di istanze in esecuzione di un gestore code è limitato a due, una è attiva e una è in standby.

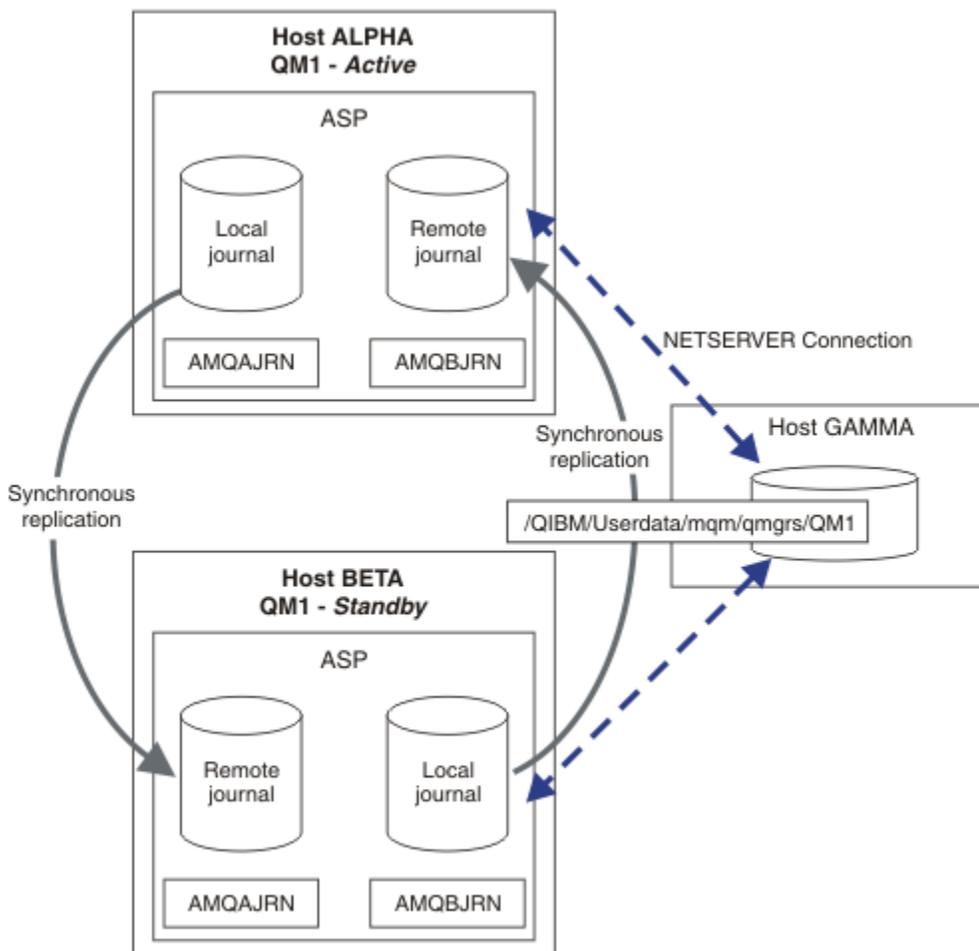


Figura 27. Eseguire il mirror di un journal del gestore code

Il giornale locale per QM1 sull'host ALPHA è denominato AMQAJRN (o più completamente, QMQM1/AMQAJRN) e su BETA il giornale è QMQM1/AMQBJRN. Ogni journal locale viene replicato sui journal remoti su tutte le altre istanze del gestore code. Se il gestore code è configurato con due istanze, un journal locale viene replicato su un journal remoto.

### Replica del giornale remoto \*SYNC o \*ASYN

I giornali IBM i vengono sottoposti a mirroring utilizzando uno dei due (\*SYNC) o asincrono (\*ASYN) registrazione su giornale; consultare [Gestione giornale remoto](#).

La modalità di replica in [Figura 27 a pagina 391](#) è \*SYNC, non \*ASYN. \*ASYN è più veloce, ma se si verifica un errore quando lo stato del giornale remoto è \*ASYNPEND, il giornale locale e quello remoto non sono congruenti. Il giornale remoto deve essere in linea con il giornale locale. Se si sceglie \*SYNC, il sistema locale attende il giornale remoto prima di ritornare da una chiamata che richiede una scrittura completata. I giornali locali e remoti generalmente rimangono coerenti tra loro. Solo se l'operazione \*SYNC impiega più tempo del tempo indicato<sup>1</sup>, e la registrazione su giornale remota è disattivata, i giornali non vengono sincronizzati. Un errore viene registrato nella coda messaggi di giornale e in QSYSOPR. Il gestore code rileva questo messaggio, scrive un errore nel log degli errori del gestore code e disattiva la replica remota del journal del gestore code. L'istanza del gestore code attiva viene ripristinata senza la registrazione del journal remoto su questo journal. Quando il server remoto è di nuovo disponibile, è necessario riattivare manualmente la replica del giornale remoto sincrona. I giornali vengono quindi risincronizzati.

<sup>1</sup> Il tempo designato è 60 secondi su IBM i 5 e nell'intervallo compreso tra 1 e 3600 secondi su IBM i 6.1 in poi.

Un problema con la configurazione \*SYNC / \*SYNC illustrata in [Figura 27 a pagina 391](#) è il modo in cui l'istanza del gestore code in standby su BETA assume controllo. Non appena l'istanza del gestore code su BETA scrive il primo messaggio persistente, tenta di aggiornare il journal remoto su ALPHA. Se la causa del passaggio del controllo da ALPHA a BETA è stato l'errore di ALPHA e ALPHA è ancora inattivo, la registrazione su giornale remota in ALPHA ha esito negativo. BETA attende la risposta di ALPHA, quindi disattiva la registrazione su giornale remota e riprende l'elaborazione dei messaggi solo con la registrazione su giornale locale. BETA deve attendere un po' per rilevare che ALPHA è inattivo, causando un periodo di inattività.

La scelta tra l'impostazione della registrazione su giornale remota su \*SYNC o su \*ASYNC è un trade - off. [Tabella 24 a pagina 392](#) riepiloga i compromessi tra l'utilizzo di \*SYNC e la registrazione su giornale \*ASYNC tra una coppia di gestori code:

<i>Tabella 24. Opzioni di registrazione su giornale remoto</i>			
<b>Attivo</b>	<b>Standby</b>	<b>*SYNC</b>	<b>*ASYNC</b>
<b>*SYNC</b>		<ol style="list-style-type: none"> <li>1. Commutazione e failover coerenti</li> <li>2. L'istanza standby non viene ripresa immediatamente dopo il failover.</li> <li>3. La registrazione su giornale remota deve essere sempre disponibile</li> <li>4. Le prestazioni dei gestori code dipendono dalla registrazione su giornale remota</li> </ol>	<ol style="list-style-type: none"> <li>1. Commutazione e failover coerenti</li> <li>2. La registrazione su giornale remota deve essere commutata in *SYNC quando il server standby è disponibile</li> <li>3. La registrazione su giornale remota deve rimanere disponibile dopo che è stata riavviata</li> <li>4. Le prestazioni dei gestori code dipendono dalla registrazione su giornale remota</li> </ol>
<b>*ASYNC</b>		<ol style="list-style-type: none"> <li>1. Non è una combinazione sensata</li> </ol>	<ol style="list-style-type: none"> <li>1. Alcuni messaggi potrebbero essere persi o duplicati dopo un failover o una commutazione</li> <li>2. Non è necessario che l'istanza in standby sia sempre disponibile perché l'istanza attiva possa continuare senza ritardo.</li> <li>3. Le prestazioni non dipendono dalla registrazione su giornale remota</li> </ol>

#### **\*SYNC / \*SYNC**

L'istanza del gestore code attiva utilizza la registrazione su giornale \*SYNC e quando l'istanza del gestore code in standby viene avviata, tenta immediatamente di utilizzare la registrazione su giornale \*SYNC .

1. Il journal remoto è transazionalmente congruente con il journal locale del gestore code attivo. Se il gestore code viene passato all'istanza in standby, può riprendere immediatamente. L'istanza in standby normalmente riprende senza alcuna perdita o duplicazione di messaggi. I messaggi vengono persi o duplicati solo se la registrazione su giornale remoto non è riuscita dall'ultimo punto di controllo e il gestore code precedentemente attivo non può essere riavviato.
2. Se il gestore code esegue il failover sull'istanza in standby, potrebbe non essere in grado di avviarsi immediatamente. L'istanza del gestore code in standby è attivata con la registrazione su giornale \*SYNC . La causa del failover potrebbe impedire la registrazione su giornale remota sul server che ospita l'istanza in standby. Il gestore code attende il rilevamento del problema prima di elaborare eventuali messaggi persistenti. Un errore viene registrato nella coda messaggi di giornale e in QSYSOPR. Il gestore code rileva questo messaggio, scrive un errore nel log degli errori del gestore code e disattiva la replica remota del journal del gestore code. L'istanza del gestore code attiva

viene ripristinata senza la registrazione del journal remoto su questo journal. Quando il server remoto è di nuovo disponibile, è necessario riattivare manualmente la replica del giornale remoto sincrona. I giornali vengono quindi risincronizzati.

3. Il server su cui viene duplicato il giornale remoto deve essere sempre disponibile per gestire il giornale remoto. Il journal remoto viene generalmente replicato sullo stesso server su cui è presente il gestore code in standby. Il server potrebbe diventare non disponibile. Un errore viene registrato nella coda messaggi di giornale e in QSYSOPR. Il gestore code rileva questo messaggio, scrive un errore nel log degli errori del gestore code e disattiva la replica remota del journal del gestore code. L'istanza del gestore code attiva viene ripristinata senza la registrazione del journal remoto su questo journal. Quando il server remoto è di nuovo disponibile, è necessario riattivare manualmente la replica del giornale remoto sincrona. I giornali vengono quindi risincronizzati.
4. La registrazione su giornale remota è più lenta della registrazione su giornale locale e sostanzialmente più lenta se i server sono separati da una grande distanza. Il gestore code deve attendere il journaling remoto, che riduce le prestazioni del gestore code.

La configurazione \*SYNC / \*SYNC tra una coppia di server ha lo svantaggio di un ritardo nella ripresa dell'istanza standby dopo il failover. La configurazione \*SYNC / \*ASYNCRON non ha questo problema.

\*SYNC / \*ASYNCRON non garantisce alcuna perdita di messaggi dopo la commutazione o il failover, purché sia disponibile un giornale remoto. Se si desidera ridurre il rischio di perdita di messaggi dopo il failover o la commutazione, sono disponibili due opzioni. Arrestare l'istanza attiva se il giornale remoto diventa inattivo oppure creare giornali remoti su più di un server.

#### **\*SYNC / \*ASYNCRON**

L'istanza del gestore code attiva utilizza il journal \*SYNC e quando viene avviata l'istanza del gestore code in standby, utilizza il journal \*ASYNCRON . Poco dopo che il server su cui si trova la nuova istanza standby diventa disponibile, l'operatore di sistema deve commutare il giornale remoto sull'istanza attiva in \*SYNC. Quando l'operatore commuta la registrazione su giornale remota da \*ASYNCRON a \*SYNC , l'istanza attiva si interrompe se lo stato del giornale remoto è \*ASYNCRONPEND. L'istanza del gestore code attivo attende che le voci del journal rimanenti vengano trasferite al journal remoto. Quando il giornale remoto è stato sincronizzato con il giornale locale, il nuovo standby è nuovamente congruente con la nuova istanza attiva. Dal punto di vista della gestione dei gestori code a più istanze, in una configurazione \*SYNC / \*ASYNCRON l'operatore di sistema IBM i ha un'attività aggiuntiva. L'operatore deve commutare la registrazione su giornale remota in \*SYNC oltre a riavviare l'istanza del gestore code in errore.

1. Il journal remoto è transazionalmente congruente con il journal locale del gestore code attivo. Se l'istanza del gestore code attivo viene commutata o se viene eseguito il failover sull'istanza in standby, l'istanza in standby può quindi riprendere immediatamente. L'istanza in standby normalmente riprende senza alcuna perdita o duplicazione di messaggi. I messaggi vengono persi o duplicati solo se la registrazione su giornale remoto non è riuscita dall'ultimo punto di controllo e il gestore code precedentemente attivo non può essere riavviato.
2. L'operatore di sistema deve commutare il giornale remoto da \*ASYNCRON a \*SYNC poco dopo che il sistema che ospita l'istanza attiva diventa nuovamente disponibile. L'operatore potrebbe attendere il recupero del giornale remoto prima di passare il giornale remoto a \*SYNC. In alternativa, l'operatore potrebbe commutare immediatamente l'istanza remota in \*SYNC e forzare l'istanza attiva ad attendere che il giornale dell'istanza in standby si riattivi. Quando la registrazione su giornale remota è impostata su \*SYNC, l'istanza in standby è generalmente congruente con l'istanza attiva. I messaggi vengono persi o duplicati solo se la registrazione su giornale remoto non è riuscita dall'ultimo punto di controllo e il gestore code precedentemente attivo non può essere riavviato.
3. Quando la configurazione è stata ripristinata da una commutazione o da un failover, il server su cui risiede il giornale remoto deve essere sempre disponibile.

Scegliere \*SYNC / \*ASYNCRON quando si vuole che il gestore code in standby riprenda rapidamente dopo un failover. È necessario ripristinare manualmente l'impostazione del giornale remoto su \*SYNC sulla nuova istanza attiva. La configurazione \*SYNC / \*ASYNCRON corrisponde al modello normale di gestione di una coppia di gestori code a più istanze. Dopo che un'istanza ha avuto esito negativo, c'è un periodo di

tempo prima che l'istanza standby venga riavviata, durante il quale l'istanza attiva non può eseguire il failover.

#### **\*ASYNC / \*ASYNC**

Entrambi i server che ospitano i gestori code attivi e in standby sono configurati per utilizzare la registrazione su giornale remota \*ASYNC .

1. Quando si verifica la commutazione o il failover, il gestore code continua con il journal sul nuovo server. Il giornale potrebbe non essere sincronizzato quando si verifica la commutazione o il failover. Di conseguenza, i messaggi potrebbero essere persi o duplicati.
2. L'istanza attiva viene eseguita, anche se il server che ospita il gestore code in standby non è disponibile. Il giornale locale viene replicato in modo asincrono con il server standby quando è disponibile.
3. Le prestazioni del gestore code locale non sono influenzate dalla registrazione su giornale remota.

Scegliere \*ASYNC / \*ASYNC se le prestazioni sono il requisito principale e si è pronti a perdere o duplicare alcuni messaggi dopo il failover o la commutazione.

#### **\*ASYNC / \*SYNC**

Non esiste alcun motivo per utilizzare questa combinazione di opzioni.

### **Attivazione del gestore code da un journal remoto**

I journal vengono replicati in modo sincrono o asincrono. Il giornale remoto potrebbe non essere attivo oppure potrebbe essere in linea con il giornale locale. Il giornale remoto potrebbe essere in fase di recupero, anche se è replicato in modo sincrono, perché potrebbe essere stato recentemente attivato. Le regole che il gestore code applica allo stato del journal remoto che utilizza durante l'avvio sono le seguenti.

1. L'avvio standby ha esito negativo se deve essere riprodotto dal giornale remoto sul giornale standby e lo stato del giornale è \*FAILED o \*INACTPEND.
2. Quando si avvia l'attivazione dello standby, lo stato del giornale remoto sullo standby deve essere \*ACTIVE o \*INACTIVE. Se lo stato è \*INACTIVE, è possibile che l'attivazione abbia esito negativo, se non tutti i dati del giornale sono stati replicati.

L'errore si verifica se i dati del gestore code sul file system di rete hanno un record di checkpoint più recente di quello presente nel journal remoto. È improbabile che l'errore si verifichi, purché il giornale remoto sia attivato entro l'intervallo massimo predefinito di 30 minuti tra i punti di controllo. Se il gestore code in standby non legge un record di checkpoint più recente dal file system, non viene avviato.

È possibile scegliere: attendere che il journal locale sul server attivo possa essere ripristinato oppure avviare a freddo il gestore code in standby. Se si sceglie l'avvio a freddo, il gestore code viene avviato senza dati journal e si basa sulla congruenza e sulla completezza dei dati del gestore code nel file system.

**Nota:** Se si avvia a freddo un gestore code, si corre il rischio di perdere o duplicare i messaggi dopo l'ultimo checkpoint. Le transazioni del messaggio sono state scritte nel journal, ma alcune delle transazioni potrebbero non essere state scritte nei dati del gestore code nel file system. Quando si avvia a freddo un gestore code, viene avviato un nuovo journal e le transazioni non scritte nei dati del gestore code nel file system vengono perse.

3. L'attivazione del gestore code in standby attende che lo stato del giornale remoto sul database standby cambi da \*ASYNCPEND o \*SYNCPEND a \*ASYNC o \*SYNC. I messaggi vengono scritti periodicamente nella registrazione lavori del controller di esecuzione.

**Nota:** In questo caso, l'attivazione è in attesa sul journal remoto locale per il gestore code standby che si sta attivando. Il gestore code attende anche un periodo di tempo prima di continuare senza un journal remoto. Attende quando tenta di scrivere in modo sincrono sul giornale remoto (o sui giornali) e il giornale non è disponibile.

4. L'attivazione si arresta se lo stato del giornale viene modificato in \*FAILED o \*INACTPEND.

I nomi e gli stati dei journal locali e remoti da utilizzare nell'attivazione vengono scritti nel log degli errori del gestore code.

## Creazione di un gestore code a più istanze utilizzando il mirroring del journal e NetServer su IBM i

Creare un gestore code a più istanza da eseguire su due server IBM i . I dati del gestore code vengono memorizzati su un terzo server IBM i utilizzando NetServer. Il journal del gestore code viene sottoposto a mirroring tra i due server utilizzando il journal remoto. Il comando **ADDQMJRN** viene utilizzato per semplificare la creazione dei giornali remoti.

### Prima di iniziare

1. L'attività richiede tre server IBM i . Installare IBM MQ su due di essi, ALPHA e BETA nell'esempio. il prodotto deve essere almeno IBM WebSphere MQ 7.0.1 Fix Pack 1.
2. Il terzo server è un server IBM i , connesso da NetServer a ALPHA e BETA. Viene utilizzato per condividere i dati del gestore code. Non è necessario disporre di un'installazione IBM MQ . È utile installare IBM MQ sul server come passo temporaneo, per impostare le autorizzazioni e le directory del gestore code.
3. Accertarsi che il profilo utente QMQM abbia la stessa password su tutti e tre i server.
4. Installare IBM i NetServer; consultare [i5/OS NetServer](#).

### Informazioni su questa attività

Effettuare le seguenti operazioni per creare la configurazione mostrata in [Figura 28 a pagina 398](#). I dati del gestore code sono connessi utilizzando IBM i NetServer.

- Creare le connessioni da ALPHA e BETA alla condivisione di directory su GAMMA che deve memorizzare i dati del gestore code. L'attività imposta anche le autorizzazioni, i profili utente e le parole d'ordine necessari.
- Aggiungere le voci del database relazionale (RDBE) ai sistemi IBM i che eseguirà le istanze del gestore code. Le voci RDBE vengono utilizzate per collegarsi ai sistemi IBM i utilizzati per la registrazione su giornale remota.
- Creare il gestore code QM1 sul server IBM i , ALPHA.
- Aggiungere le informazioni di controllo del gestore code per QM1 sull'altro IBM i server, BETA.
- Creare journal remoti su entrambi i server IBM i per entrambe le istanze del gestore code. Ogni gestore code scrive nel journal locale. Il giornale locale viene replicato sul giornale remoto. Il comando, **ADDQMJRN** semplifica l'aggiunta dei giornali e delle connessioni.
- Avviare il gestore code, consentendo un'istanza in standby.

### Procedura

1. Eseguire l'attività, "Creazione di una condivisione di rete per i dati del gestore code utilizzando NetServer su IBM i" a pagina 382.

Di conseguenza, ALPHA e BETA hanno una condivisione, /QNTC/GAMMA/WMQ, che punta a /QIBM/UserData/mqm/qmgrs su GAMMA. I profili utente QMQM e QMQMADM hanno le autorizzazioni necessarie e QMQM ha parole d'ordine corrispondenti su tutti e tre i sistemi.

2. Aggiungere le voci del database relazionale (RDBE) ai sistemi IBM i che ospiteranno le istanze del gestore code.
  - a) Su ALPHA creare la connessione a BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) Su BETA creare le connessioni a ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Creare il gestore code QM1 su ALPHA, salvando i dati del gestore code su GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Il percorso, utilizza NetServer per creare i dati del gestore code.

4. Esegui su ALPHA. Il comando aggiunge un giornale remoto su BETA per.

```
ADDMQMJRN MQMNAME(QM1) RMTJRN RDB(BETA)
```

crea le voci di giornale nel relativo giornale locale su ALPHA quando l'istanza attiva è su ALPHA. Il giornale locale su ALPHA viene replicato sul giornale remoto su BETA.

5. Utilizzare il comando, per esaminare i dati di configurazione IBM MQ creati da ALPHA.

Le informazioni sono necessarie nel prossimo passo.

In questo esempio, la seguente configurazione viene creata in ALPHA per:

```
Name=QM1  
Prefix=/QIBM/UserData/mqm  
Library=QM1  
Directory=QM1  
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. Creare un'istanza del gestore code di QM1 su BETA utilizzando il comando. Eseguire il seguente comando su BETA per modificare le informazioni di controllo del gestore code in BETA.

```
ADDMQMINF MQMNAME(QM1)  
PREFIX('/QIBM/UserData/mqm')  
MQMDIR(QM1)  
MQMLIB(QM1)  
DATAPATH('/QNTC/GAMMA/WMQ /QM1 ')
```

**Suggerimento:** Copiare e incollare le informazioni di configurazione. La stanza del gestore code è la stessa su ALPHA e BETA.

7. Esegui su BETA. Il comando aggiunge un giornale locale su BETA e un giornale remoto su ALPHA per.

```
ADDMQMJRN MQMNAME(QM1) RMTJRN RDB(ALPHA)
```

crea voci di giornale nel relativo giornale locale su BETA quando l'istanza attiva è su BETA. Il giornale locale su BETA viene replicato sul giornale remoto su ALPHA.

**Nota:** In alternativa, è possibile impostare la registrazione su giornale remota da BETA a ALPHA utilizzando la registrazione su giornale asincrona.

Utilizzare questo comando per impostare la registrazione su giornale asincrona da BETA a ALPHA, invece del comando nel passo [“7” a pagina 396](#).

```
ADDMQMJRN MQMNAME(QM1) RMTJRN RDB(ALPHA) RMTJRN DLV(*ASYNCR)
```

Se il server o la registrazione su giornale su ALPHA è l'origine dell'errore, BETA viene avviato senza attendere che le nuove voci di giornale vengano replicate su ALPHA.

Passare la modalità di replica a \*SYNC, utilizzando il comando, quando ALPHA è di nuovo in linea.

Utilizzare le informazioni contenute in [“Configurazione giornale di mirroring per ASP su IBM i” a pagina 389](#) per decidere se eseguire il mirroring dei giornali in modo sincrono, asincrono o una combinazione

di entrambi. Il valore predefinito è di replicare in modo sincrono, con un periodo di attesa di 60 secondi per una risposta dal giornale remoto.

8. Verificare che i giornali su ALPHA e BETA siano abilitati e che lo stato della replica del giornale remoto sia abilitato.

a) Su ALPHA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) Su BETA:

```
WRKMQMJRN MQMNAME(QM1)
```

9. Avviare le istanze del gestore code su ALPHA e BETA.

a) Avviare la prima istanza su ALPHA, rendendola l'istanza attiva. Abilitazione del passaggio a un'istanza standby.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Avviare la seconda istanza su BETA, rendendola l'istanza standby.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

## Risultati

Utilizzare per controllare lo stato del gestore code:

1. Lo stato dell'istanza del gestore code su ALPHA deve essere.
2. Lo stato dell'istanza del gestore code su BETA dovrebbe essere.

## Esempio

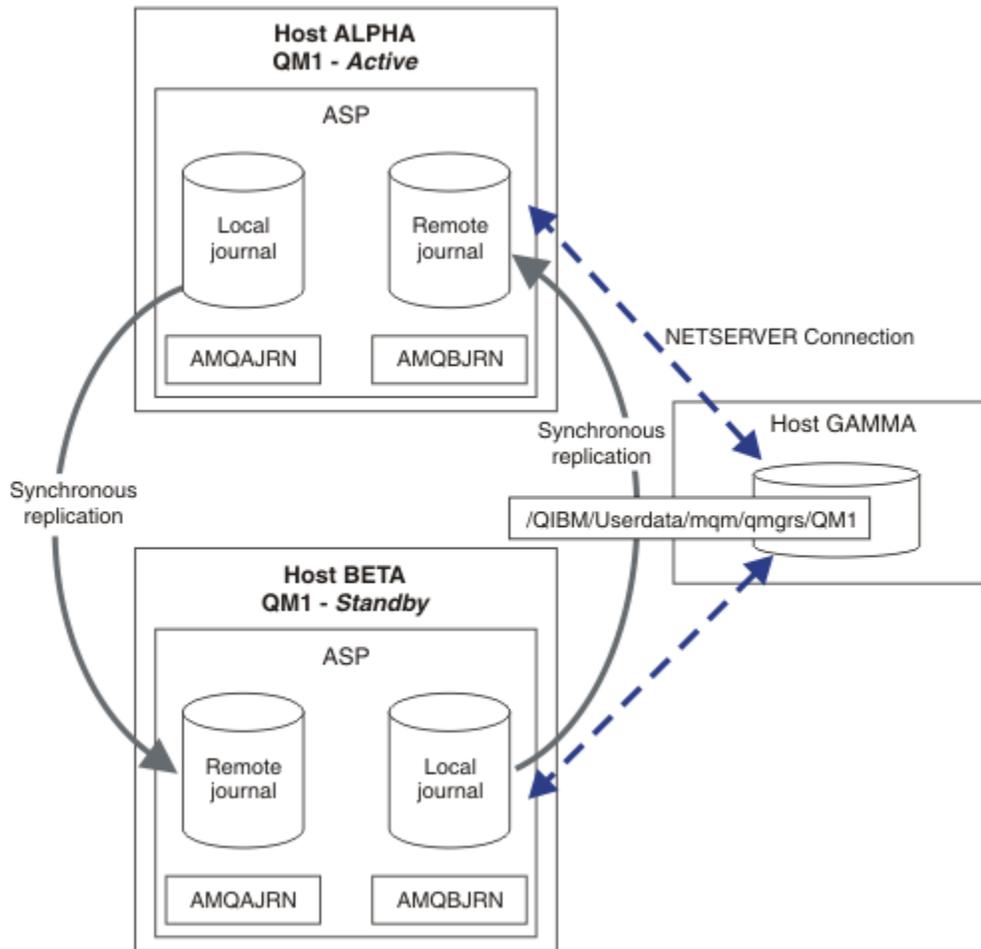


Figura 28. Configurazione giornale con mirroring

## Operazioni successive

- Verificare che le istanze attive e in standby si spostino automaticamente. È possibile eseguire i programmi di esempio di alta disponibilità di esempio per verificare lo switch over; consultare [Programmi di esempio alta disponibilità](#). I programmi di esempio sono client 'C'. È possibile eseguirli da una piattaforma Windows o Unix.

1. Avviare i programmi di esempio ad alta disponibilità.
2. Su ALPHA, terminare il gestore code che richiede il passaggio su:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Verificare che l'istanza su BETA sia attiva.
4. Riavvia su ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Esaminare le configurazioni alternative di alta disponibilità:
  1. Utilizzare NetServer per inserire i dati del gestore code su un server Windows .

2. Invece di utilizzare la registrazione su giornale remota per eseguire il mirroring del journal del gestore code, memorizzare il journal su un ASP indipendente. Utilizzare il cluster IBM i per trasferire l'ASP indipendente da ALPHA a BETA.

### **IBM i** *Conversione di un gestore code a istanza singola in un gestore code a più istanze utilizzando NetServer e il mirroring del journal su IBM i*

Convertire un gestore code a istanza singola in gestore code a più istanze. Spostare i dati del gestore code in una condivisione di rete connessa da NetServer. Eseguire il mirroring del journal del gestore code su un secondo server IBM i utilizzando il journal remoto.

#### **Prima di iniziare**

1. L'attività richiede tre server IBM i . L'installazione di IBM MQ esistente, sul server ALPHA nell'esempio, deve essere almeno IBM WebSphere MQ 7.0.1 Fix Pack 1. ALPHA sta eseguendo un gestore code denominato QM1 nell'esempio.
2. Installare IBM MQ sul secondo server IBM i , BETA nell'esempio.
3. Il terzo server è un server IBM i , connesso da NetServer a ALPHA e BETA. Viene utilizzato per condividere i dati del gestore code. Non è necessario disporre di un'installazione IBM MQ . È utile installare IBM MQ sul server come passo temporaneo, per impostare le autorizzazioni e le directory del gestore code.
4. Accertarsi che il profilo utente QMQM abbia la stessa password su tutti e tre i server.
5. Installare IBM i NetServer; consultare [i5/OS NetServer](#).

#### **Informazioni su questa attività**

Effettuare le seguenti operazioni per convertire un gestore code a istanza singola nel gestore code a più istanze mostrato in [Figura 29](#) a pagina 403. Il gestore code a istanza singola viene eliminato nell'attività e quindi ricreato, memorizzando i dati del gestore code sulla condivisione di rete connessa da NetServer. Questa procedura è più affidabile dello spostamento delle directory e dei file del gestore code nella condivisione di rete utilizzando il comando **CPY** .

- Creare le connessioni da ALPHA e BETA alla condivisione di directory su GAMMA che deve memorizzare i dati del gestore code. L'attività imposta anche le autorizzazioni, i profili utente e le parole d'ordine necessari.
- Aggiungere le voci del database relazionale (RDBE) ai sistemi IBM i che eseguirà le istanze del gestore code. Le voci RDBE vengono utilizzate per collegarsi ai sistemi IBM i utilizzati per la registrazione su giornale remota.
- Salvare le definizioni e i log del gestore code, arrestare il gestore code ed eliminarlo.
- Ricreare il gestore code, memorizzando i dati del gestore code sulla condivisione di rete su GAMMA.
- Aggiungere la seconda istanza del gestore code all'altro server.
- Creare journal remoti su entrambi i server IBM i per entrambe le istanze del gestore code. Ogni gestore code scrive nel journal locale. Il giornale locale viene replicato sul giornale remoto. Il comando, **ADDQMJRN** semplifica l'aggiunta dei giornali e delle connessioni.
- Avviare il gestore code, consentendo un'istanza in standby.

#### **Nota:**

Nel passo [“4” a pagina 400](#) dell'attività, si elimina il gestore code a istanza singola, QM1. L'eliminazione del gestore code elimina tutti i messaggi persistenti nelle code. Per questo motivo, completare l'elaborazione di tutti i messaggi memorizzati dal gestore code prima di convertire il gestore code. Se non è possibile elaborare tutti i messaggi, eseguire il backup della libreria del gestore code prima del passo [“4” a pagina 400](#). Ripristinare la libreria del gestore code dopo il passo [“5” a pagina 400](#).

#### **Nota:**

Nel passo [“5” a pagina 400](#) dell'attività, creare nuovamente QM1. Anche se il gestore code ha lo stesso nome, ha un identificativo del gestore code diverso. Il cluster del gestore code utilizza l'identificativo

del gestore code. Per eliminare e creare nuovamente un gestore code in un cluster, è necessario prima rimuovere il gestore code dal cluster; fare riferimento a [Rimozione di un gestore code da un cluster: metodo alternativo](#) o [Rimozione di un gestore code da un cluster](#). Una volta ricreato il gestore code, aggiungerlo al cluster. Sebbene abbia lo stesso nome di prima, sembra essere un nuovo gestore code per gli altri gestori code nel cluster.

## Procedura

1. Eseguire l'attività, [“Creazione di una condivisione di rete per i dati del gestore code utilizzando NetServer su IBM i”](#) a pagina 382.

Di conseguenza, ALPHA e BETA hanno una condivisione, /QNTC/GAMMA/WMQ, che punta a /QIBM/UserData/mqm/qmgrs su GAMMA. I profili utente QMQM e QMQMADM hanno le autorizzazioni necessarie e QMQM ha parole d'ordine corrispondenti su tutti e tre i sistemi.

2. Aggiungere le voci del database relazionale (RDBE) ai sistemi IBM i che ospiteranno le istanze del gestore code.
  - a) Su ALPHA creare la connessione a BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) Su BETA creare le connessioni a ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Creare gli script che ricreano gli oggetti gestore code.

```
QSAVEQMGR LCLQMGRNAM(QM1) FILENAME(' *CURLIB/QMQSC(QM1) ')  
OUTPUT(*REPLACE) MAKEAUTH(*YES) AUTHFN(' *CURLIB/QMAUT(QM1) ')
```

4. Arrestare il gestore code ed eliminarlo.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*YES) TIMEOUT(15)  
DLTMQM MQMNAME(QM1)
```

5. Creare il gestore code QM1 su ALPHA, salvando i dati del gestore code su GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Il percorso, utilizza NetServer per creare i dati del gestore code.

6. Ricreare gli oggetti del gestore code per QM1 dalle definizioni salvate.

```
STRMQMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. Applicare le autorizzazioni dalle informazioni salvate.

- a) Compilare il programma di autorizzazione salvato.

```
CRTCLPGM PGM(*CURLIB/QM1) SRCFILE(*CURLIB/QMAUT)  
SRCMBR(QM1) REPLACE(*YES)
```

- b) Eseguire il programma per applicare le autorizzazioni.

```
CALL PGM(*CURLIB/QM1)
```

- c) Aggiornare le informazioni di sicurezza per QM1.

```
RFRMQMAUT MQMNAME(QM1)
```

8. Eseguì su ALPHA. Il comando aggiunge un giornale remoto su BETA per.

```
ADDQMQRN MQMNAME(QM1) RMTJRNDRB(BETA)
```

crea le voci di giornale nel relativo giornale locale su ALPHA quando l'istanza attiva è su ALPHA. Il giornale locale su ALPHA viene replicato sul giornale remoto su BETA.

9. Utilizzare il comando, per esaminare i dati di configurazione IBM MQ creati da ALPHA.

Le informazioni sono necessarie nel prossimo passo.

In questo esempio, la seguente configurazione viene creata in ALPHA per:

```
Name=QM1  
Prefix=/QIBM/UserData/mqm  
Library=QM1  
Directory=QM1  
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. Creare un'istanza del gestore code di QM1 su BETA utilizzando il comando. Eseguire il seguente comando su BETA per modificare le informazioni di controllo del gestore code in BETA.

```
ADDQMINF MQMNAME(QM1)  
PREFIX('/QIBM/UserData/mqm')  
MQMDIR(QM1)  
MQMLIB(QM1)  
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

**Suggerimento:** Copiare e incollare le informazioni di configurazione. La stanza del gestore code è la stessa su ALPHA e BETA.

11. Eseguì su BETA. Il comando aggiunge un giornale locale su BETA e un giornale remoto su ALPHA per.

```
ADDQMQRN MQMNAME(QM1) RMTJRNDRB(ALPHA)
```

crea voci di giornale nel relativo giornale locale su BETA quando l'istanza attiva è su BETA. Il giornale locale su BETA viene replicato sul giornale remoto su ALPHA.

**Nota:** In alternativa, è possibile impostare la registrazione su giornale remota da BETA a ALPHA utilizzando la registrazione su giornale asincrona.

Utilizzare questo comando per impostare la registrazione su giornale asincrona da BETA a ALPHA, invece del comando nel passo [“7”](#) a pagina 396.

```
ADDQMQRN MQMNAME (QM1) RMTJRNDRB (ALPHA) RMTJRNDRB (*ASYNCR)
```

Se il server o la registrazione su giornale su ALPHA è l'origine dell'errore, BETA viene avviato senza attendere che le nuove voci di giornale vengano replicate su ALPHA.

Passare la modalità di replica a \*SYNCR, utilizzando il comando, quando ALPHA è di nuovo in linea.

Utilizzare le informazioni contenute in [“Configurazione giornale di mirroring per ASP su IBM i”](#) a pagina 389 per decidere se eseguire il mirroring dei giornali in modo sincrono, asincrono o una combinazione di entrambi. Il valore predefinito è di replicare in modo sincrono, con un periodo di attesa di 60 secondi per una risposta dal giornale remoto.

12. Verificare che i giornali su ALPHA e BETA siano abilitati e che lo stato della replica del giornale remoto sia abilitato.

a) Su ALPHA:

```
WRKMQMJRN MQMNAME(QM1)
```

b) Su BETA:

```
WRKMQMJRN MQMNAME(QM1)
```

13. Avviare le istanze del gestore code su ALPHA e BETA.

a) Avviare la prima istanza su ALPHA, rendendola l'istanza attiva. Abilitazione del passaggio a un'istanza standby.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

b) Avviare la seconda istanza su BETA, rendendola l'istanza standby.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

## Risultati

Utilizzare per controllare lo stato del gestore code:

1. Lo stato dell'istanza del gestore code su ALPHA deve essere.
2. Lo stato dell'istanza del gestore code su BETA dovrebbe essere.

## Esempio

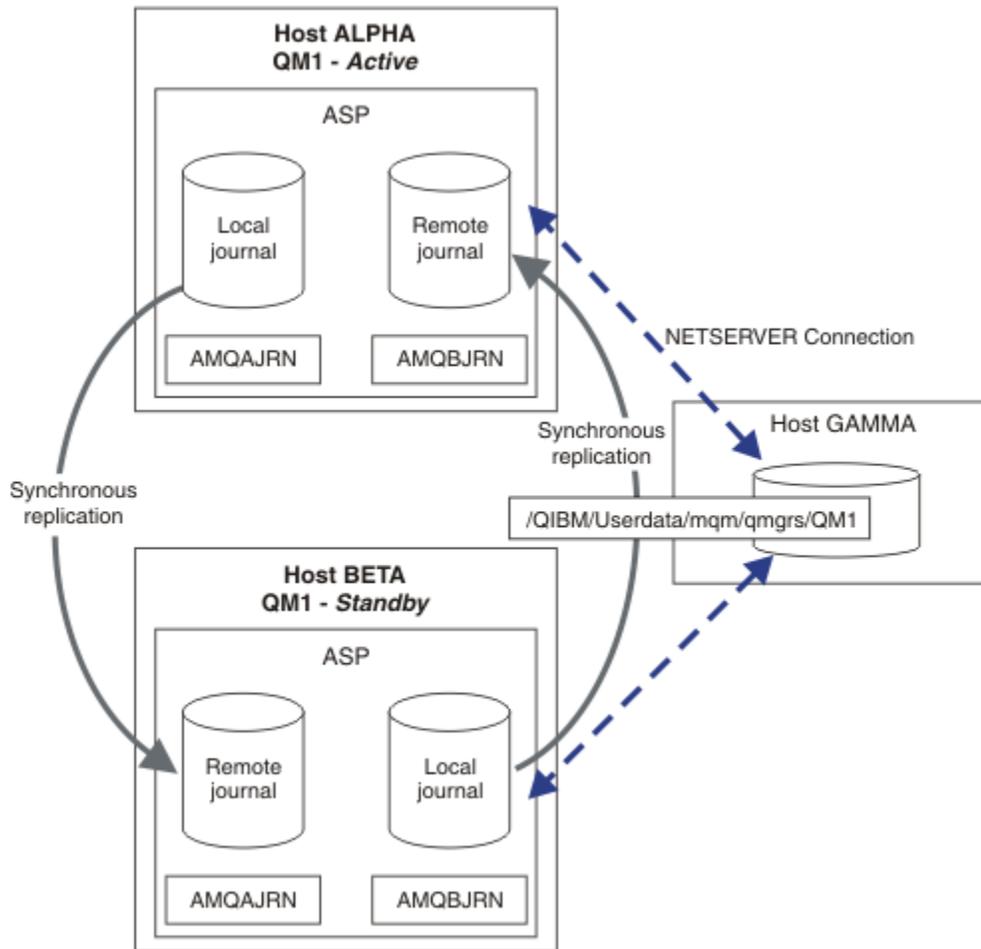


Figura 29. Configurazione giornale con mirroring

## Operazioni successive

- Verificare che le istanze attive e in standby si spostino automaticamente. È possibile eseguire i programmi di esempio di alta disponibilità di esempio per verificare lo switch over; consultare [Programmi di esempio alta disponibilità](#). I programmi di esempio sono client 'C'. È possibile eseguirli da una piattaforma Windows o Unix.

1. Avviare i programmi di esempio ad alta disponibilità.
2. Su ALPHA, terminare il gestore code che richiede il passaggio su:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Verificare che l'istanza su BETA sia attiva.
4. Riavvia su ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Esaminare le configurazioni alternative di alta disponibilità:
  1. Utilizzare NetServer per inserire i dati del gestore code su un server Windows .

- Invece di utilizzare la registrazione su giornale remota per eseguire il mirroring del journal del gestore code, memorizzare il journal su un ASP indipendente. Utilizzare il cluster IBM i per trasferire l'ASP indipendente da ALPHA a BETA.

### **IBM i** **Configurazione giornale ASP indipendente commutata su IBM i**

Non è necessario replicare un journal ASP indipendente per creare una configurazione del gestore code a più istanze. È necessario automatizzare un mezzo per trasferire l'ASP indipendente dal gestore code attivo al gestore code in standby. Esistono soluzioni alternative di alta disponibilità possibili utilizzando un ASP indipendente, non tutte richiedono l'utilizzo di un gestore code a più istanze.

Quando si utilizza un ASP indipendente non è necessario eseguire il mirroring del journal del gestore code. Se è stata installata la gestione del cluster e i server che ospitano le istanze del gestore code si trovano nello stesso gruppo di risorse cluster, il journal del gestore code può essere trasferito automaticamente su un altro server entro una breve distanza dal server attivo, se l'host che esegue l'istanza attiva ha esito negativo. È anche possibile trasferire il giornale manualmente, come parte di uno switch pianificato, oppure è possibile scrivere una procedura di comando per trasferire l'ASP indipendente in modo programmatico.

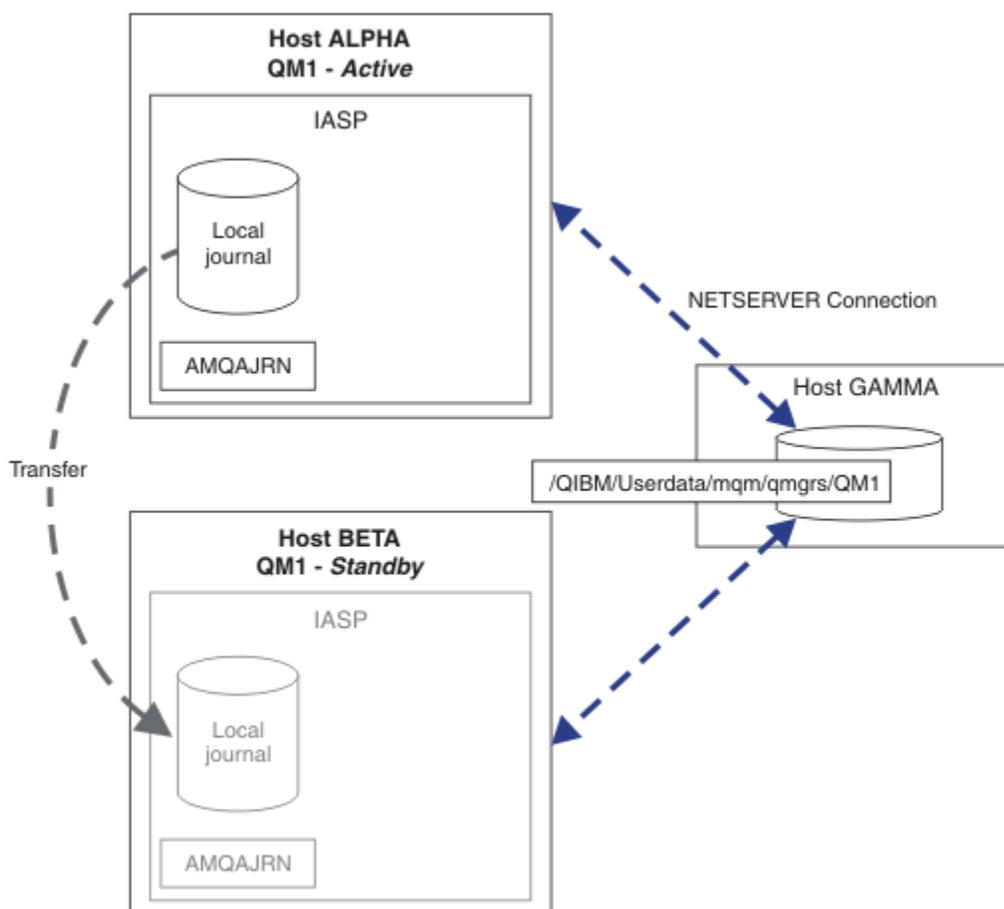


Figura 30. Trasferire un journal del gestore code utilizzando un ASP indipendente

Per l'operazione del gestore code a più istanze, i dati del gestore code devono essere memorizzati su un file system condiviso. Il file system può essere ospitato su una varietà di piattaforme differenti. Non è possibile memorizzare i dati del gestore code a più istanze su un ASP o un ASP indipendente.

Il file system condiviso esegue due ruoli nella configurazione: gli stessi dati del gestore code sono condivisi tra tutte le istanze del gestore code. Il file system deve avere un robusto protocollo di blocco che garantisce che una sola istanza del gestore code abbia accesso ai dati del gestore code una volta avviato. Se il gestore code ha esito negativo o le comunicazioni con il file server si interrompono, il file system

deve rilasciare il blocco sui dati del gestore code conservati dall'istanza attiva che non comunica più con il file system. L'istanza del gestore code in standby può ottenere l'accesso in lettura / scrittura ai dati del gestore code. Il protocollo del file system deve essere conforme a una serie di regole per funzionare correttamente con gestori code a più istanze; consultare [“Componenti di una soluzione alta disponibilità su IBM i” a pagina 381.](#)

Il meccanismo di blocco serializza il comando di avvio del gestore code e controlla quale istanza del gestore code è attiva. Una volta che un gestore code diventa attivo, ricrea le proprie code dal journal locale che l'utente, o il cluster HA, ha trasferito al server standby. I client ricollegabili in attesa di riconnessione allo stesso gestore code vengono riconnessi e viene eseguito il backout di tutte le transazioni in corso. Le applicazioni configurate per essere avviate come servizi del gestore code vengono avviate.

È necessario verificare che il journal locale dell'istanza del gestore code attivo non riuscirà sull'ASP indipendente venga trasferito al server su cui è presente l'istanza del gestore code in standby appena attivata, configurando il gestore risorse cluster o trasferendo manualmente l'ASP indipendente. L'utilizzo di ASP indipendenti non preclude la configurazione dei giornali remoti e del mirroring, se si decide di utilizzare l'ASP indipendente per il backup e il ripristino di emergenza e utilizzare il mirroring del journal remoto per la configurazione del gestore code a più istanze.

Se si è scelto di utilizzare un ASP indipendente, è possibile considerare configurazioni alternative ad alta disponibilità. Lo sfondo di queste soluzioni è descritto in [“ASP indipendenti e alta disponibilità” a pagina 408.](#)

1. Invece di utilizzare i gestori code a più istanze, installare e configurare un gestore code a istanza singola interamente su un ASP indipendente e utilizzare i servizi di alta disponibilità IBM i per eseguire il failover del gestore code. È probabilmente necessario aumentare la soluzione con un monitor del gestore code per rilevare se il gestore code ha avuto esito negativo indipendentemente dal server. Questa è la base della soluzione fornita in *Supportpac MC41: Configurazione di IBM MQ per iSeries per l'alta disponibilità.*
2. Utilizzare gli ASP indipendenti e XSM (cross site mirroring) per eseguire il mirroring dell'ASP indipendente piuttosto che commutare l'ASP indipendente sul bus locale. Ciò estende l'intervallo geografico della soluzione ASP indipendente fino a quando il tempo impiegato per scrivere i record di registrazione su una lunga distanza lo consente.

#### Creazione di un gestore code a più istanze utilizzando un ASP indipendente e NetServer su IBM i

Creare un gestore code a più istanza da eseguire su due server IBM i . I dati del gestore code vengono memorizzati su un server IBM i utilizzando NetServer. Il journal del gestore code è memorizzato su un ASP indipendente. Utilizzare il cluster IBM i o una procedura manuale per trasferire l'ASP indipendente contenente il journal del gestore code sull'altro server IBM i .

### Prima di iniziare

1. L'attività richiede tre server IBM i . Installare IBM MQ su due di essi, ALPHA e BETA nell'esempio. il prodotto deve essere almeno IBM WebSphere MQ 7.0.1 Fix Pack 1.
2. Il terzo server è un server IBM i , connesso da NetServer a ALPHA e BETA. Viene utilizzato per condividere i dati del gestore code. Non è necessario disporre di un'installazione IBM MQ . È utile installare IBM MQ sul server come passo temporaneo, per impostare le autorizzazioni e le directory del gestore code.
3. Accertarsi che il profilo utente QMQM abbia la stessa password su tutti e tre i server.
4. Installare IBM i NetServer; consultare [i5/OS NetServer.](#)
5. Creare le procedure per trasferire l'ASP indipendente dal gestore code in errore al gestore code in standby che sta subendo il controllo. È possibile che alcune delle tecniche contenute in *SupportPac MC41: Configurazione di IBM MQ per iSeries per l'alta disponibilità* siano utili nella progettazione delle procedure di trasferimento ASP indipendente.

## Informazioni su questa attività

Effettuare le seguenti operazioni per creare la configurazione mostrata in [Figura 31 a pagina 407](#). I dati del gestore code sono connessi utilizzando IBM i NetServer.

- Creare le connessioni da ALPHA e BETA alla condivisione di directory su GAMMA che deve memorizzare i dati del gestore code. L'attività imposta anche le autorizzazioni, i profili utente e le parole d'ordine necessari.
- Creare il gestore code QM1 sul server IBM i , ALPHA.
- Aggiungere le informazioni di controllo del gestore code per QM1 sull'altro IBM i server, BETA.
- Avviare il gestore code, consentendo un'istanza in standby.

## Procedura

1. Eseguire l'attività, [“Creazione di una condivisione di rete per i dati del gestore code utilizzando NetServer su IBM i”](#) a pagina 382.

Di conseguenza, ALPHA e BETA hanno una condivisione, /QNTC/GAMMA/WMQ, che punta a /QIBM/UserData/mqm/qmgrs su GAMMA. I profili utente QMQM e QMQMADM hanno le autorizzazioni necessarie e QMQM ha parole d'ordine corrispondenti su tutti e tre i sistemi.

2. Creare il gestore code QM1 su ALPHA, salvando i dati del gestore code su GAMMA.

```
CRTMQM MQMNAME(QM1) UDLSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ ')
```

Il percorso, utilizza NetServer per creare i dati del gestore code.

3. Utilizzare il comando, per esaminare i dati di configurazione IBM MQ creati da ALPHA.

Le informazioni sono necessarie nel prossimo passo.

In questo esempio, la seguente configurazione viene creata in ALPHA per:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMQM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. Creare un'istanza del gestore code di QM1 su BETA utilizzando il comando. Eseguire il seguente comando su BETA per modificare le informazioni di controllo del gestore code in BETA.

```
ADDQMINF MQMNAME(QM1)
PREFIX(' /QIBM/UserData/mqm ')
MQMDIR(QM1)
MQMLIB(QMQM1)
DATAPATH(' /QNTC/GAMMA/WMQ /QM1 ')
```

**Suggerimento:** Copiare e incollare le informazioni di configurazione. La stanza del gestore code è la stessa su ALPHA e BETA.

5. Avviare le istanze del gestore code su ALPHA e BETA.
  - a) Avviare la prima istanza su ALPHA, rendendola l'istanza attiva. Abilitazione del passaggio a un'istanza standby.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) Avviare la seconda istanza su BETA, rendendola l'istanza standby.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

## Risultati

Utilizzare per controllare lo stato del gestore code:

1. Lo stato dell'istanza del gestore code su ALPHA deve essere.
2. Lo stato dell'istanza del gestore code su BETA dovrebbe essere.

## Esempio

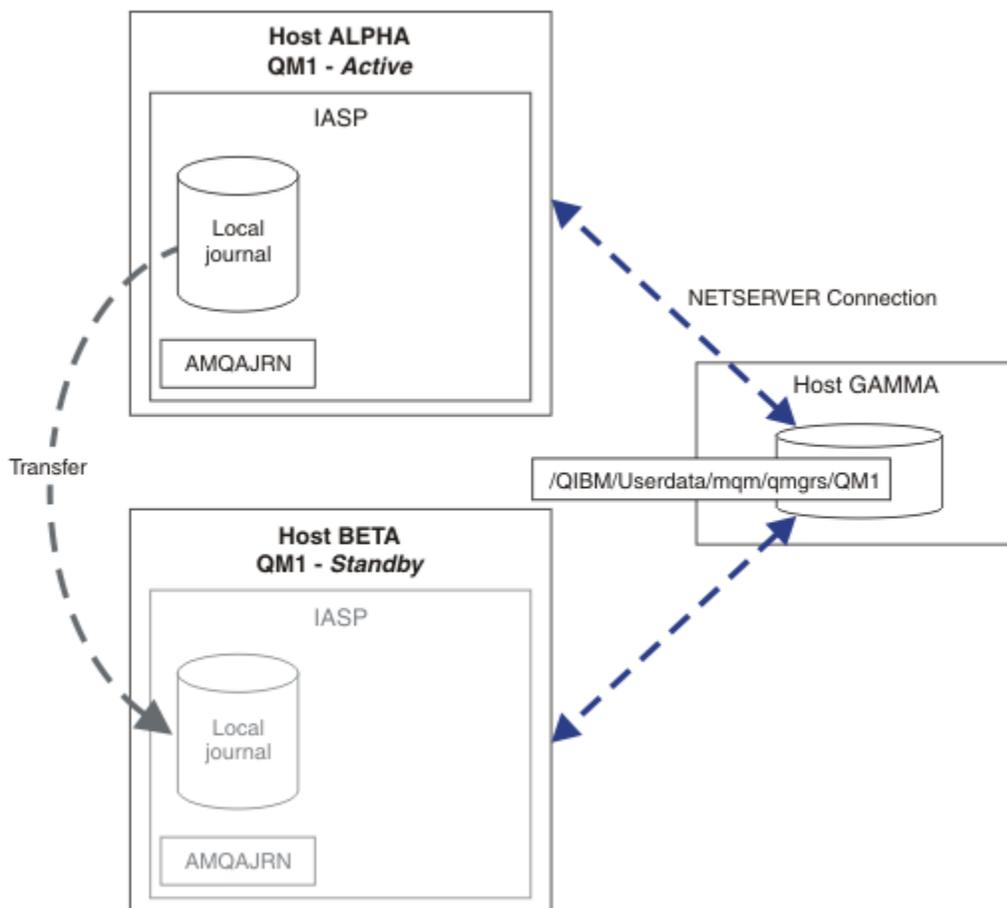


Figura 31. Trasferire un journal del gestore code utilizzando un'ASP indipendente

## Operazioni successive

- Verificare che le istanze attive e in standby si spostino automaticamente. È possibile eseguire i programmi di esempio di alta disponibilità di esempio per verificare lo switch over; consultare Programmi di esempio alta disponibilità. I programmi di esempio sono client 'C'. È possibile eseguirli da una piattaforma Windows o Unix.

1. Avviare i programmi di esempio ad alta disponibilità.
2. Su ALPHA, terminare il gestore code che richiede il passaggio su:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Verificare che l'istanza su BETA sia attiva.
4. Riavvia su ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Esaminare le configurazioni alternative di alta disponibilità:
  1. Utilizzare NetServer per posizionare i dati del gestore code su un server IBM i .
  2. Invece di utilizzare un ASP indipendente per trasferire il journal del gestore code sul server standby, utilizzare la registrazione su giornale remota per eseguire il mirroring del journal sul server standby.

### IBM i ASP indipendenti e alta disponibilità

Gli ASP indipendenti consentono alle applicazioni e ai dati di essere spostati tra i server. La flessibilità degli ASP indipendenti significa che sono la base per alcune soluzioni di alta disponibilità IBM i . Nel considerare se utilizzare un ASP o un ASP indipendente per il journal del gestore code, è necessario considerare un'altra configurazione ad alta disponibilità basata su ASP indipendenti.

Gli ASP (auxiliary storage pools) sono un blocco di creazione dell'architettura di IBM i . Le unità disco sono raggruppate insieme per formare un singolo ASP. Posizionando gli oggetti in diversi ASP, è possibile proteggere i dati in un ASP da eventuali errori del disco in un altro ASP.

Ogni server IBM i ha almeno un ASP *di base* , noto come ASP di sistema. È designato come ASP1e talvolta è noto come \*SYSBAS. È possibile configurare fino a 31 ASP *utente* di base aggiuntivi che non sono distinguibili dall'ASP di sistema dal punto di vista dell'applicazione, poiché condividono lo stesso spazio dei nomi. Utilizzando più ASP di base per distribuire le applicazioni su più dischi è possibile migliorare le prestazioni e ridurre il tempo di recupero. L'utilizzo di più ASP di base può anche fornire un certo grado di isolamento rispetto al malfunzionamento del disco, ma non migliora l'affidabilità in generale.

Gli ASP indipendenti sono un tipo speciale di ASP. Sono spesso chiamati lotti dischi indipendenti. I lotti dischi indipendenti sono componenti chiave dell'alta disponibilità IBM i . È possibile memorizzare i dati e le applicazioni che si considerano indipendenti dal sistema corrente a cui sono collegati sulle unità di memoria disco. È possibile configurare ASP indipendenti commutabili o non commutabili. Da una prospettiva di disponibilità, generalmente si è interessati solo agli ASP indipendenti commutabili, che possono essere trasferiti automaticamente da server a server. Come risultato è possibile spostare le applicazioni e i dati sull'ASP indipendente da server a server.

A differenza degli ASP utente di base, gli ASP indipendenti non condividono lo stesso spazio nome dell'ASP di sistema. Le applicazioni che gestiscono gli ASP utente richiedono modifiche per gestire un ASP indipendente. È necessario verificare che il software e il software di terze parti utilizzati funzionino in un ambiente ASP indipendente.

Quando l'ASP indipendente è collegato ad un altro server, lo spazio nome dell'ASP indipendente deve essere combinato con lo spazio nome del sistema ASP. Questo processo è denominato *attivazione* dell'ASP indipendente. È possibile attivare un ASP indipendente senza eseguire l'IPL del server. Il supporto cluster è richiesto per trasferire automaticamente gli ASP indipendenti da un server ad un altro.

## Costruire soluzioni affidabili con ASP indipendenti

La registrazione su giornale in un ASP indipendente, piuttosto che la registrazione su giornale in un ASP e l'utilizzo della replica del journal, fornisce un mezzo alternativo per fornire al gestore code in standby una copia del journal locale dall'istanza del gestore code in errore. Per trasferire automaticamente l'ASP indipendente ad un altro server, è necessario avere installato e configurato il supporto cluster. Esistono diverse soluzioni di alta disponibilità per gli ASP indipendenti basate sul supporto del cluster e sul mirroring del disco di basso livello, che è possibile combinare o sostituire utilizzando gestori code a più istanze.

Il seguente elenco descrive i componenti necessari per creare una soluzione affidabile basata su ASP indipendenti.

### Registrazione su giornale

I gestori code e altre applicazioni utilizzano i journal locali per scrivere i dati persistenti in modo sicuro sul disco per proteggerli dalla perdita di dati in memoria a causa di un errore del server. Questa è a volte definita consistenza point - in - time. Non garantisce la coerenza di più aggiornamenti che si verificano in un periodo di tempo.

## **Controllo del commit**

Utilizzando le transazioni globali, è possibile coordinare gli aggiornamenti di messaggi e database in modo che i dati scritti nel journal siano congruenti. Fornisce coerenza per un periodo di tempo utilizzando un protocollo di commit a due fasi.

## **Disco commutato**

I dischi commutati vengono gestiti dal CRG unità in un cluster HA. CRG commuta automaticamente gli ASP indipendenti in un nuovo server in caso di interruzione non pianificata. I CRG sono geograficamente limitati all'estensione del bus IO locale.

Configurando il giornale locale su un ASP indipendente commutabile, è possibile trasferirlo su un server differente e riprendere l'elaborazione dei messaggi. Nessuna modifica ai messaggi persistenti effettuata senza il controllo del punto di sincronizzazione o di cui è stato eseguito il commit con il controllo del punto di sincronizzazione, viene persa, a meno che l'ASP indipendente non abbia esito negativo.

Se si utilizza sia la registrazione su giornale che il controllo del commit su ASP indipendenti commutabili, è possibile trasferire i journal del database e i journal del gestore code su un server differente e riprendere l'elaborazione delle transazioni senza alcuna perdita di congruenza o transazioni sottoposte a commit.

## **XSM (Cross - site mirroring)**

XSM esegue il mirroring dell'ASP indipendente primario in un ASP indipendente secondario geograficamente remoto su una rete TCP/IP e trasferisce il controllo automaticamente in caso di errore. È possibile configurare un mirror sincrono o asincrono. Il mirroring sincrono riduce le prestazioni del gestore code poiché i dati vengono sottoposti a mirroring prima del completamento delle operazioni di scrittura sul sistema di produzione, ma garantisce che l'ASP indipendente secondario sia aggiornato. Mentre se si utilizza il mirroring asincrono, non è possibile garantire che l'ASP indipendente secondario sia aggiornato. Il mirroring asincrono mantiene la congruenza dell'ASP indipendente secondario.

Ci sono tre tecnologie XSM.

### **Mirroring geografico**

Il mirroring geografico è un'estensione del cluster, che consente di commutare gli ASP indipendenti in un'ampia area. Ha entrambe le modalità sincrone e asincrone. È possibile garantire l'alta disponibilità solo in modalità sincrone, ma la separazione degli ASP indipendenti potrebbe influire troppo sulle prestazioni. È possibile combinare il mirroring geografico con il disco commutato per fornire l'alta disponibilità localmente e il ripristino di emergenza in remoto.

### **Mirroring metro**

Il mirroring metro è un servizio a livello di unità che fornisce il mirroring sincrono locale veloce su distanze più lunghe rispetto al bus locale. È possibile combinarlo con un gestore code a più istanze per fornire l'alta disponibilità del gestore code e, avendo due copie dell'ASP indipendente, l'elevata disponibilità del journal del gestore code.

### **Mirroring globale**

Il mirroring globale è un servizio a livello unità che fornisce il mirroring asincrono ed è adatto per il backup e il ripristino di emergenza su distanze più lunghe, ma non è una scelta normale per l'alta disponibilità, perché mantiene solo la congruenza del punto nel tempo piuttosto che la valuta.

I principali punti di decisione da considerare sono:

### **ASP o ASP indipendente?**

Non è necessario eseguire un cluster IBM i HA per utilizzare gestori code a più istanze. È possibile scegliere gli ASP indipendenti, se si stanno già utilizzando gli ASP indipendenti, oppure si hanno requisiti di disponibilità per altre applicazioni che richiedono gli ASP indipendenti. Potrebbe essere utile combinare gli ASP indipendenti con i gestori code a più istanze per sostituire il monitoraggio del gestore code come mezzo per rilevare gli errori del gestore code.

### **Disponibilità?**

Qual è l'obiettivo del tempo di recupero (RTO)? Se si richiede l'aspetto di un comportamento quasi ininterrotto, allora quale soluzione ha il tempo di recupero più veloce?

### Disponibilità giornale?

Come eliminare il giornale come un singolo punto di errore. È possibile adottare una soluzione hardware, utilizzando periferiche RAID 1 o superiori, oppure combinare o utilizzare una soluzione software utilizzando i journal di replica o il mirroring del disco.

### Distanza?

La distanza tra le istanze del gestore code attive e in standby. Gli utenti possono tollerare il degrado delle prestazioni della replica in modo sincrono su distanze superiori a circa 250 metri?

### Competenze?

C'è del lavoro da fare per automatizzare le attività amministrative coinvolte nella manutenzione ed esercizio regolare della soluzione. Le competenze richieste per eseguire l'automazione sono diverse per le soluzioni basate su ASP e ASP indipendenti.

## **Eliminazione di un gestore code a più istanze su IBM i**

Prima di eliminare un gestore code a più istanze, arrestare la registrazione su giornale remota e rimuovere istanze del gestore code.

### Prima di iniziare

1. In questo esempio, due istanze del gestore code QM1 sono definite sui server ALPHA e BETA. ALPHA è l'istanza attiva e BETA è lo standby. I dati del gestore code associati al gestore code QM1 sono memorizzati sul IBM i server GAMMA, utilizzando NetServer. Consultare [“Creazione di un gestore code a più istanze utilizzando il mirroring del journal e NetServer su IBM i”](#) a pagina 395.
2. ALPHA e BETA devono essere collegati in modo che tutti i giornali remoti definiti possano essere eliminati da IBM MQ.
3. Verificare che sia possibile accedere alla directory /QNTC e alla condivisione file della directory del server utilizzando i comandi di sistema **EDTF** o **WRKLNK**

### Informazioni su questa attività

Prima di eliminare un gestore code a più istanze da un server utilizzando il comando **DLTMQM**, rimuovere tutte le istanze del gestore code su altri server utilizzando il comando **RMVMQMINF**.

Quando si rimuove un'istanza del gestore code utilizzando il comando **RMVMQMINF**, i journal locali e remoti con prefisso AMQe associati all'istanza, vengono eliminati. Vengono eliminate anche le informazioni di configurazione relative all'istanza del gestore code, locale per il server.

Non eseguire il comando **RMVMQMINF** sul server che contiene l'istanza rimanente del gestore code. In questo modo si impedisce a **DLTMQM** di funzionare correttamente.

Eliminare il gestore code utilizzando il comando **DLTMQM**. I dati del gestore code vengono rimossi dalla condivisione di rete. I giornali locali e remoti con prefisso AMQ e associati all'istanza vengono eliminati. **DLTMQM** elimina anche le informazioni di configurazione sull'istanza del gestore code, locale per il server.

Nell'esempio, esistono solo due istanze del gestore code. IBM MQ supporta una configurazione a più istanze in esecuzione che ha un'istanza del gestore code attiva e un'istanza in standby. Se sono state create ulteriori istanze del gestore code da utilizzare nelle configurazioni in esecuzione, rimuoverle, utilizzando il comando **RMVMQMINF**, prima di eliminare l'istanza rimanente.

### Procedura

1. Eseguire il comando **CHGMQMJRN RMTJRNSTS (\*INACTIVE)** su ciascun server per rendere inattivo il journaling remoto tra le istanze del gestore code.
  - a) Su ALPHA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRN RDB('BETA') RMTJRNSTS(*INACTIVE)
```

- b) Su BETA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. Eseguire il comando **ENDMQM** su ALPHA, l'istanza del gestore code attivo, per arrestare entrambe le istanze di QM1.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

3. Eseguire il comando **RMVMQMINF** su ALPHA per rimuovere le risorse del gestore code per l'istanza da ALPHA e BETA.

```
RMVMQMINF MQMNAME(QM1)
```

**RMVMQMINF** rimuove le informazioni di configurazione del gestore code per QM1 da ALPHA. Se il nome del journal ha come prefisso AMQ, elimina il journal locale associato a QM1 da ALPHA. Se il nome del giornale è preceduto dal prefisso AMQ ed è stato creato un giornale remoto, rimuove anche il giornale remoto da BETA.

4. Eseguire il comando **DLTMQM** su BETA per eliminare QM1.

```
DLTMQM MQMNAME(QM1)
```

**DLTMQM** elimina i dati del gestore code dalla condivisione di rete su GAMMA. Rimuove le informazioni di configurazione del gestore code per QM1 da BETA. Se il nome del giornale ha come prefisso AMQ, elimina il giornale locale associato a QM1 da BETA. Se il nome del giornale è preceduto da AMQ ed è stato creato un giornale remoto, rimuove anche il giornale remoto da ALPHA.

## Risultati

**DLTMQM** e **RMVMQMINF** eliminano i giornali locali e remoti creati da **CRTMQM** e **ADDMQJRN**. I comandi cancellano anche i ricevitori di giornale. I giornali e i ricevitori di giornale devono seguire la convenzione di denominazione dei nomi che iniziano con AMQ. **DLTMQM** e **RMVMQMINF** rimuovono gli oggetti del gestore code, i dati del gestore code e le informazioni di configurazione del gestore code da `mqs.ini`.

## Operazioni successive

Un approccio alternativo consiste nell'immettere i seguenti comandi dopo la disattivazione della registrazione su giornale nel passo [“1” a pagina 410](#) e prima di terminare le istanze del gestore code. Oppure, se non è stata seguita la convenzione di denominazione, è necessario cancellare i giornali e i ricevitori di giornale per nome.

1. Su ALPHA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. Su BETA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Dopo aver eliminato i journal, continuare con il resto dei passi.

## **Esecuzione del backup di un gestore code a più istanze su IBM i**

La procedura mostra come eseguire il backup degli oggetti del gestore code sul server locale e i dati del gestore code sul server di file di rete. Adattare l'esempio per eseguire il backup dei dati per altri gestori code.

## Prima di iniziare

In questo esempio, i dati del gestore code associati con il gestore code QM1 vengono memorizzati sul IBM i server denominato GAMMA, utilizzando NetServer. Consultare [“Creazione di un gestore code a più istanze utilizzando il mirroring del journal e NetServer su IBM i”](#) a pagina 395. IBM MQ è installato sui server ALPHA e BETA. Il gestore code, QM1, è configurato su ALPHA e BETA.

## Informazioni su questa attività

IBM i non supporta il salvataggio dei dati da una directory remota. Salvare i dati del gestore code su un file system remoto utilizzando le procedure di backup locali sul server del filesystem. In questa attività, il file system di rete si trova su un server IBM i, GAMMA. Viene eseguito il backup dei dati del gestore code in un file di salvataggio su GAMMA.

Se il file system di rete si trovava su Windows o Linux, è possibile memorizzare i dati del gestore code in un file compresso e salvarli. Se si dispone di un sistema di backup, come Tivoli Storage Manager, utilizzarlo per eseguire il backup dei dati del gestore code.

## Procedura

1. Creare un file di salvataggio su ALPHA per la libreria del gestore code associato a QM1.

Utilizzare il nome della libreria del gestore code per denominare il file di salvataggio.

```
CRTSAVF FILE(QGPL/QMQM1)
```

2. Salvare la libreria del gestore code nel file di salvataggio su ALPHA.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

3. Creare un file di salvataggio per la directory dei dati del gestore code su GAMMA.

Utilizzare il nome gestore code per denominare il file di salvataggio.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

4. Salvare la copia dei dati del gestore code dalla directory locale su GAMMA.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

## Comandi per impostare gestori code a più istanze

IBM MQ dispone di dei comandi per semplificare la configurazione della replica del journal, l'aggiunta di nuove istanze del gestore code e la configurazione dei gestori code per utilizzare l'ASP indipendente.

I comandi di giornale per creare e gestire i giornali locali e remoti sono:

### ADDMQMJRN

Con questo comando è possibile creare dei journal locali e remoti denominati per un'istanza del gestore code e configurare se la replica è sincrona o asincrona, qual è il timeout sincrono e se il journal remoto deve essere attivato immediatamente.

### CHGMQMJRN

Il comando modifica il timeout, lo stato e i parametri di consegna che interessano i journal di replica.

### RMVMQMJRN

Rimuove i journal denominati *remoti* da un'istanza del gestore code.

### WRKMQMJRN

Elenca lo stato dei journal locali e remoti per un'istanza del gestore code locale.

Aggiungere e gestire istanze aggiuntive del gestore code utilizzando i seguenti comandi, che modificano il file `mq5.ini`.

#### **ADDMQMINF**

Il comando usa le informazioni estratte dal file `mq5.ini` con il comando `DSPMQMINF` per aggiungere una nuova istanza del gestore code su un server IBM i differente.

#### **RMVMQMINF**

Rimuovere un'istanza del gestore code. Utilizzare questo comando per rimuovere un'istanza di un gestore code esistente o per rimuovere le informazioni di configurazione per un gestore code che è stato eliminato da un altro server.

Il comando **CRTMQM** dispone di tre parametri per assistere nella configurazione di un gestore code a più istanze,

#### **MQMDIRP (\*DFT | prefisso - indirizzario)**

Utilizzare questo parametro per selezionare un punto di montaggio associato ai dati del gestore code sulla memoria di rete.

#### **ASP (\*SYSTEM|\*ASPDEV| numero - lotto - memoria - ausiliaria)**

Specificare `*SYSTEM` o un *numero ASP (auxiliary - storage - pool - number)* per posizionare il journal del gestore code sul sistema o su un ASP utente di base. Selezionare l'opzione `*ASPDEV` e impostare un nome unità utilizzando il parametro **ASPDEV**, per posizionare il journal del gestore code su un ASP indipendente.

#### **ASPDEV (\*ASP|nome - unità)**

Specificare un *nome - unità* di un'unità ASP indipendente principale o secondaria. La selezione di `*ASP` ha lo stesso risultato della specifica di **ASP** (`*SYSTEM`).

IBM i

## **Considerazioni sulle prestazioni e sul failover del disco su IBM i**

Utilizzare diversi lotti di memoria ausiliaria per migliorare le prestazioni e l'affidabilità.

Se si utilizza un numero elevato di messaggi persistenti o messaggi di grandi dimensioni nelle applicazioni, il tempo impiegato per scrivere questi messaggi sul disco diventa un fattore significativo nelle prestazioni del sistema.

Accertarsi di disporre di un'attivazione disco sufficiente per far fronte a questa possibilità oppure considerare un ASP (Auxiliary Storage Pool) separato in cui conservare i ricevitori di giornale del gestore code.

È possibile specificare su quale ASP vengono memorizzati la libreria del gestore code e i journal quando si crea il gestore code utilizzando il parametro ASP di **CRTMQM**. Per impostazione predefinita, la libreria del gestore code e i journal e dati IFS vengono memorizzati nell'ASP di sistema.

Gli ASP consentono l'isolamento degli oggetti su una o più specifiche unità disco. Ciò può ridurre la perdita di dati a causa di un malfunzionamento del supporto disco. Nella maggior parte dei casi, vengono persi solo i dati memorizzati sulle unità disco nell'ASP interessato.

Si consiglia di archiviare la libreria del gestore code e i dati journal in ASP utente separati in quello del file system IFS root per fornire il failover e ridurre il conflitto del disco.

Per ulteriori informazioni, consultare [Backup e ripristino](#) nella documentazione di IBM i.

IBM i

## **Utilizzo di SAVLIB per salvare le librerie IBM MQ su IBM i**

Non è possibile utilizzare `SAVLIB LIB(*ALLUSR)` per salvare le librerie IBM MQ, poiché tali librerie hanno nomi che iniziano con Q.

È possibile utilizzare `SAVLIB LIB(QM*)` per salvare tutte le librerie del gestore code, ma solo se si sta utilizzando un'unità di salvataggio diversa da `*SAVF`. Per `DEV(*SAVF)`, è necessario utilizzare un comando `SAVLIB` per ogni libreria del gestore code sul sistema.

## IBM i In corso di quiesce IBM MQ for IBM i

Questa sezione spiega come disattivare (terminare correttamente) IBM MQ for IBM i.

Per disattivare IBM MQ for IBM i:

1. Collegarsi a una nuova sessione IBM MQ for IBM i interattiva, assicurandosi di non accedere ad alcun oggetto.
2. Accertarsi di avere:
  - Autorizzazione \*ALLOBJ o autorizzazione di gestione oggetto per la libreria QMQM
  - Autorizzazione sufficiente per utilizzare il comando ENDSBS
3. Avvisare tutti gli utenti che si sta per arrestare IBM MQ for IBM i.
4. Il modo in cui si procede dipende dal fatto che si desidera arrestare (disattivare) un singolo gestore code (dove potrebbero esistere altri) (consultare [“Arresto di un singolo gestore code per IBM MQ for IBM i” a pagina 414](#)) o tutti i gestori code (consultare [“Arresto di tutti i gestori code per IBM MQ for IBM i” a pagina 416](#)).
5. Arrestare il server mqweb immettendo il seguente comando in qshell:

```
/QIBM/ProdData/mqm/bin/endmqweb
```

### ENDMQM parametro ENDCCTJOB (\*YES)

Il parametro ENDMQM ENDCCTJOB (\*YES) funziona in modo diverso in IBM MQ for IBM i V6.0 e successive rispetto alle versioni precedenti.

Nelle versioni precedenti, quando si specifica ENDCCTJOB (\*YES), MQ termina forzatamente le applicazioni.

Su IBM MQ for IBM i V6.0 o successiva, quando si specifica ENDCCTJOB (\*YES), le proprie applicazioni non vengono terminate ma vengono invece scollegate dal gestore code.

Se si specifica ENDCCTJOB (\*YES) e si dispone di applicazioni che non vengono scritte per rilevare che un gestore code è in fase di chiusura, la volta successiva che viene emessa una nuova chiamata MQI, la chiamata restituirà un errore MQRC\_CONNECTION\_BROKEN (2009).

Come alternativa all'utilizzo di ENDCCTJOB (\*YES), utilizzare il parametro ENDCCTJOB (\*NO) e utilizzare l'opzione WRKMQM 22 (Gestione lavori) per terminare manualmente tutti i lavori dell'applicazione che impediranno il riavvio di un gestore code.

## IBM i Arresto di un singolo gestore code per IBM MQ for IBM i

Utilizzare queste informazioni per comprendere i tre tipi di arresto.

Nelle procedure che seguono, viene utilizzato un nome di gestore code di esempio QMgr1 e un nome di sottosistema di esempio SUBX. Sostituire questi nomi con i propri valori, se necessario.

### Chiusura pianificata

Arresto pianificato di un gestore code su IBM i

1. Prima della chiusura, eseguire:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

2. Per arrestare il gestore code, eseguire:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Se QMgr1 non termina, il canale o le applicazioni sono probabilmente occupati.

3. Se è necessario arrestare immediatamente QMgr1 , eseguire quanto riportato di seguito:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

## Arresto non pianificato

1. Per arrestare il gestore code, eseguire:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Se QMgr1 non termina, il canale o le applicazioni sono probabilmente occupati.

2. Se è necessario arrestare QMgr1 immediatamente, eseguire quanto riportato di seguito:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

## Chiusura in condizioni anomale

1. Per arrestare il gestore code, eseguire:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Se QMgr1 non termina, continuare con il passo 3 a condizione che:

- QMgr1 si trova nel proprio sottosistema oppure
  - È possibile terminare tutti i gestori code che condividono lo stesso sottosistema di QMgr1. Utilizzare la procedura di arresto non pianificato per tutti i gestori code.
2. Dopo aver eseguito tutte le operazioni della procedura per tutti i gestori code che condividono il sottosistema ( SUBX negli esempi), eseguire:

```
ENDSBS SUBX *IMMED
```

Se il completamento di questo comando non riesce, arrestare tutti i gestori code, utilizzando la procedura di chiusura non pianificata ed eseguire un IPL sulla macchina.

**Avviso:** Non utilizzare ENDJOBABN per i lavori IBM MQ che non riescono a terminare come risultato di ENDJOB o ENDSBS, a meno che non si sia pronti ad eseguire un IPL sulla macchina immediatamente dopo.

3. Avviare il sottosistema eseguendo:

```
STRSBS SUBX
```

4. Chiudere immediatamente il gestore code, eseguendo:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(10)
```

5. Riavviare il gestore code eseguendo:

```
STRMQM MQMNAME(QMgr1)
```

Se questo non riesce, e si:

- Aver riavviato la macchina eseguendo un IPL oppure

- Avere solo un singolo gestore code

Riordinare la memoria condivisa IBM MQ eseguendo:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

prima di ripetere il passo 5.

Se il riavvio del gestore code richiede più di pochi secondi, IBM MQ aggiunge i messaggi di stato in modo intermittente al log del lavoro che descrive l'avanzamento dell'avvio.

Se hai ancora problemi a riavviare il tuo gestore code, contatta il supporto IBM . Qualsiasi ulteriore azione che potrebbe essere intrapresa potrebbe danneggiare il gestore code, lasciando IBM MQ nell'impossibilità di eseguire il ripristino.

**IBM i**

## Arresto di tutti i gestori code per IBM MQ for IBM i

Utilizzare queste informazioni per comprendere i tre tipi di arresto.

Le procedure sono quasi le stesse di un singolo gestore code, ma utilizzando \*ALL invece del nome del gestore code, laddove possibile, e utilizzando un comando che utilizza ripetutamente ciascun nome di gestore code. Nelle procedure, viene utilizzato un nome di gestore code di esempio QMgr1 e un nome di sottosistema di esempio SUBX. Sostituirli con i propri.

### Chiusura pianificata

1. Un'ora prima della chiusura, eseguire:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

Ripetere questa operazione per ogni gestore code che si desidera arrestare.

2. Per arrestare il gestore code, eseguire:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Ripetere questa operazione per ogni gestore code che si desidera arrestare; è possibile eseguire comandi separati in parallelo.

Se un gestore code non termina entro un periodo di tempo ragionevole (ad esempio 10 minuti), procedere con il passo 3.

3. Per arrestare immediatamente tutti i gestori code, eseguire quanto segue:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

### Arresto non pianificato

1. Per arrestare un gestore code, eseguire:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Ripetere questa operazione per ogni gestore code che si desidera arrestare; è possibile eseguire comandi separati in parallelo.

Se i gestori code non vengono terminati, il canale o le applicazioni sono probabilmente occupati.

2. Se è necessario arrestare immediatamente i gestori code, eseguire le seguenti operazioni:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

## Chiusura in condizioni anomale

1. Per arrestare i gestori code, eseguire:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Ripetere questa operazione per ogni gestore code che si desidera arrestare; è possibile eseguire comandi separati in parallelo.

2. Terminare i sistemi secondari ( SUBX negli esempi), eseguendo:

```
ENDSBS SUBX *IMMED
```

Ripetere questa operazione per ogni sottosistema che si desidera arrestare; i comandi separati possono essere eseguiti in parallelo.

Se questo comando non viene completato, eseguire un IPL sul sistema.

**Avviso:** Non utilizzare ENDJOBABN per i lavori che non riescono a terminare come risultato di ENDJOB o ENDSBS, a meno che non si sia pronti ad eseguire un IPL sul sistema immediatamente dopo.

3. Avviare i sottosistemi eseguendo:

```
STRSBS SUBX
```

Ripetere questa operazione per ogni sottosistema che si desidera avviare.

4. Chiudere immediatamente i gestori code, eseguendo:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. Riavviare i gestori code eseguendo:

```
STRMQM MQMNAME(QMgr1)
```

Ripetere questa operazione per ogni gestore code che si desidera avviare.

Se un riavvio del gestore code impiega più di pochi secondi, IBM MQ visualizzerà i messaggi di stato in modo intermittente che descrivono in dettaglio l'avanzamento dell'avvio.

Se hai ancora problemi a riavviare un gestore code, contatta il supporto IBM . Qualsiasi ulteriore azione che si potrebbe intraprendere potrebbe danneggiare i gestori code, lasciando MQSeries o IBM MQ incapaci di eseguire il ripristino.

La gestione dei gestori code e delle risorse associate include le attività che vengono eseguite di frequente per attivare e gestire tali risorse. Scegliere il metodo che si preferisce per gestire i gestori code e le risorse associate.

IBM MQ for z/OS può essere controllata e gestita da una serie di programmi di utilità e programmi forniti con il prodotto. È possibile utilizzare i comandi IBM MQ Script (MQSC) o PCF (Programmable Command Format) per gestire IBM MQ for z/OS. Per informazioni sull'utilizzo dei comandi per IBM MQ for z/OS, consultare [“Immissione di comandi in IBM MQ for z/OS” a pagina 418](#).

IBM MQ for z/OS fornisce inoltre una serie di programmi di utilità che consentono di gestire il sistema. Per informazioni sui diversi programmi di utilità e su come utilizzarli, consultare [“I programmi di utilità IBM MQ for z/OS”](#) a pagina 426.

Per i dettagli su come amministrare IBM MQ for z/OS e le diverse attività amministrative che potrebbe essere necessario intraprendere, consultare i seguenti link:

### **Concetti correlati**

[Concetti di IBM MQ for z/OS](#)

### **Attività correlate**

[“Utilizzo degli oggetti IBM MQ locali”](#) a pagina 122

È possibile gestire oggetti IBM MQ locali per supportare programmi applicativi che utilizzano MQI (Message Queue Interface).

[“Utilizzo di oggetti IBM MQ remoti”](#) a pagina 187

È possibile gestire oggetti IBM MQ su gestori code remoti utilizzando i comandi MQSC, PCF o amministrative REST API. Prima di poter utilizzare uno di questi metodi, è necessario definire le code di trasmissione e i canali tra il gestore code locale e il gestore code remoto in modo che i comandi possano essere inviati al gestore code remoto e le risposte ricevute dal gestore code locale. In alternativa, è possibile configurare un cluster di gestore code e utilizzare gli stessi metodi di amministrazione remota.

[“Amministrazione IBM MQ”](#) a pagina 7

Per gestire i gestori code IBM MQ e le risorse associate, scegliere il metodo preferito da una serie di attività che è possibile utilizzare per attivare e gestire tali risorse.

[Pianificazione](#)

[Pianificazione dell'ambiente IBM MQ su z/OS](#)

[Configurazione](#)

[Configurazione di z/OS](#)

[Utilizzo dei programmi di utilità IBM MQ for z/OS](#)

### **Riferimenti correlati**

[“Origini da cui è possibile emettere comandi MQSC su z/OS”](#) a pagina 20

I comandi MQSC possono essere emessi da varie origini, a seconda del comando.

[Riferimento ai formati dei comandi programmabili](#)

**z/OS**

## **Immissione di comandi in IBM MQ for z/OS**

È possibile utilizzare i comandi script IBM MQ (MQSC) in modalità batch o interattiva per controllare un gestore code.

IBM MQ for z/OS supporta i comandi MQSC, che possono essere emessi dalle seguenti origini:

- La console z/OS o equivalente (come SDSF/TSO).

Quando si utilizza la console z/OS, è necessario aggiungere /cpf all'inizio di un comando, dove cpf è il prefisso del comando per il sottosistema del gestore code.

- I dataset di input di inizializzazione.
- Il programma di utilità batch fornito, CSQUTIL, che elabora un elenco di comandi in un dataset sequenziale.

Quando si utilizzano i dataset di input di inizializzazione o il programma di utilità batch fornito, non è necessario aggiungere /cpf all'inizio di un comando.

- Un'applicazione debitamente autorizzata, inviando un comando come messaggio alla coda di input del comando. L'applicazione può essere una delle seguenti:
  - Un programma di regione batch
  - Un'applicazione CICS
  - Un'applicazione IMS
  - Un'applicazione TSO

- Un programma di applicazione o un programma di utilità su un altro sistema IBM MQ

Tabella 26 a pagina 422 riepiloga i comandi MQSC e le origini da cui possono essere emessi.

La maggior parte delle funzionalità di questi comandi è disponibile in modo conveniente dai pannelli di controllo e dalle operazioni IBM MQ for z/OS .

Le modifiche apportate alle definizioni delle risorse di un gestore code utilizzando i comandi (direttamente o indirettamente) vengono conservate nei riavvii del sottosistema IBM MQ .

IBM MQ for z/OS supporta anche comandi PCF (Programmable Command Format). Ciò semplifica la creazione di applicazioni per l'amministrazione di IBM MQ. I comandi MQSC sono in formato di testo leggibile, mentre PCF consente alle applicazioni di creare richieste e leggere le risposte senza dover analizzare le stringhe di testo. Come i comandi MQSC, le applicazioni emettono comandi PCF inviandoli come messaggi alla coda di immissione comandi. Per ulteriori informazioni sull'utilizzo dei comandi PCF e per i dettagli dei comandi, consultare la documentazione [Riferimento per i formati dei comandi programmabili](#) .

## Definizioni private e globali su IBM MQ for z/OS

Quando si definisce un oggetto su IBM MQ for z/OS, è possibile scegliere se si desidera condividere tale definizione con altri gestori code (una definizione *globale* ) o se la definizione dell'oggetto deve essere utilizzata solo da un gestore code (una definizione *privata* ). Viene chiamato *disposizione* dell'oggetto.

### Definizione globale

Se il gestore code appartiene a un gruppo di condivisione code, è possibile scegliere di condividere qualsiasi definizione di oggetto eseguita con gli altri membri del gruppo. Ciò significa che è necessario definire un oggetto una sola volta, riducendo il numero totale di definizioni richieste per l'intero sistema.

Le definizioni degli oggetti globali sono conservate in un *repository condiviso* (un Db2 database condiviso) e sono disponibili per tutti i gestori code nel gruppo di condivisione code. Questi oggetti hanno una disposizione di GRUPPO.

### Definizione privata

Se si desidera creare una definizione oggetto richiesta solo da un gestore code o se il gestore code non è un membro di un gruppo di condivisione code, è possibile creare definizioni oggetto non condivise con altri membri di un gruppo di condivisione code.

Le definizioni di oggetti privati vengono conservate nella serie di pagine zero del gestore code di definizione. Questi oggetti hanno una disposizione QMGR.

È possibile creare definizioni private per tutti i tipi di oggetti IBM MQ ad eccezione delle strutture CF (ossia, canali, elenchi nomi, definizioni di processo, code, gestori code, definizioni di classe di memoria e oggetti delle informazioni di autenticazione) e definizioni globali per tutti i tipi di oggetti ad eccezione dei gestori code.

IBM MQ copia automaticamente la definizione di un oggetto gruppo nella serie di pagine zero di ogni gestore code che lo utilizza. Se lo si desidera, è possibile modificare temporaneamente la copia della definizione e IBM MQ consente di aggiornare le copie della serie di pagine dalla copia del repository, se necessario.

IBM MQ tenta sempre di aggiornare le copie della serie di pagine dalla copia del repository all'avvio (per i comandi del canale, questa operazione viene eseguita quando l'iniziatore del canale viene riavviato) o se l'oggetto del gruppo viene modificato.

**Nota:** La copia della definizione viene aggiornata dalla definizione del gruppo, solo se la definizione del gruppo è stata modificata dopo la creazione della copia della definizione.

Ciò garantisce che le copie della serie di pagine riflettano la versione sul repository, incluse le modifiche apportate quando il gestore code era inattivo. Le copie vengono aggiornate generando i comandi DEFINE REPLACE, pertanto vi sono circostanze in cui l'aggiornamento non viene eseguito, ad esempio:

- Se una copia della coda è aperta, un aggiornamento che modifica l'utilizzo della coda non riesce.
- Se una copia di una coda contiene messaggi, l'aggiornamento che elimina la coda non riesce.
- Se una copia di una coda richiede ALTER con FORCE per modificarla.

In queste circostanze, l'aggiornamento non viene eseguito su quella copia, ma viene eseguito sulle copie su tutti gli altri gestori code.

Se il gestore code viene arrestato e quindi riavviato in modo autonomo, tutte le copie locali degli oggetti vengono eliminate, a meno che, ad esempio, la coda non disponga di messaggi associati.

Esiste una terza disposizione oggetto che si applica solo alle code locali. Ciò consente di creare code condivise. La definizione di una coda condivisa è contenuta nel repository condiviso ed è disponibile per tutti i gestori code nel gruppo di condivisione code. Inoltre, i messaggi su una coda condivisa sono disponibili anche per tutti i gestori code nel gruppo di condivisione code. Ciò è descritto in [Code condivise e gruppi di condivisione code](#). Le code condivise hanno una disposizione oggetto SHARED.

La seguente tabella riassume l'effetto delle opzioni di disposizione degli oggetti per i gestori code avviati in modo autonomo e come membro di un gruppo di condivisione code.

Disposizione	Gestore code autonomo	Membro di un gruppo di condivisione code
QMGR	Definizione oggetto conservata nella serie di pagine zero.	Definizione oggetto conservata nella serie di pagine zero.
GRUPPO	non sono consentiti.	Definizione oggetto contenuta nel repository condiviso. Copia locale conservata nella serie di pagine zero di ogni gestore code del gruppo.
CONDIVISO	non sono consentiti.	Definizione di coda conservata nel repository condiviso. Messaggi disponibili per qualsiasi gestore code nel gruppo.

## Manipolazione delle definizioni globali

Se si desidera modificare la definizione di un oggetto contenuto nel repository condiviso, è necessario specificare se si desidera modificare la versione nel repository o la copia locale nella serie di pagine zero. Utilizzare la disposizione dell'oggetto come parte del comando per effettuare questa operazione.

### Indirizzamento di comandi a gestori code differenti su z/OS

È possibile utilizzare l' *ambito del comando* per controllare su quale gestore code viene eseguito il comando.

È possibile scegliere di eseguire un comando sul gestore code in cui viene immesso o su un gestore code differente nel gruppo di condivisione code. È anche possibile scegliere di immettere un comando particolare in parallelo su tutti i gestori code in un gruppo di condivisione code. Ciò è possibile sia per i comandi MQSC che per i comandi PCF.

Ciò è determinato dall' *ambito del comando*. L'ambito del comando viene utilizzato con la disposizione degli oggetti per stabilire quale versione di un oggetto si desidera gestire.

Ad esempio, è possibile modificare alcuni degli attributi di un oggetto, la cui definizione è contenuta nel repository condiviso.

- È possibile modificare la versione solo su un gestore code e non apportare alcuna modifica alla versione del repository o a quelle utilizzate da altri gestori code.
- È possibile modificare la versione nel repository condiviso per gli utenti futuri, ma lasciare invariate le copie esistenti.

- È possibile che si desideri modificare la versione nel repository condiviso, ma anche che le modifiche vengano riflesse immediatamente su tutti i gestori code nel gruppo di condivisione code che detengono una copia dell'oggetto nella serie di pagine zero.

Utilizzare l'ambito del comando per specificare se il comando viene eseguito su questo gestore code, su un altro gestore code o su tutti i gestori code. Utilizzare la disposizione dell'oggetto per specificare se l'oggetto che si sta manipolando si trova nel repository condiviso (un oggetto del gruppo) o è una copia locale sulla serie di pagine zero (un oggetto del gestore code).

Non è necessario specificare l'ambito del comando e la disposizione dell'oggetto per gestire una coda condivisa poiché ogni gestore code nel gruppo di condivisione code gestisce la coda condivisa come una coda singola.

## Riepilogo dei comandi per IBM MQ for z/OS

Utilizzare questo argomento come riferimento dei comandi MQSC e PCF principali.

Tabella 25 a pagina 421 riepiloga i comandi MQSC e PCF disponibili su IBM MQ for z/OS per modificare, definire, eliminare e visualizzare oggetti IBM MQ .

*Tabella 25. Riepilogo dei comandi MQSC e PCF principali per tipo di oggetto*

<b>Comando MQSC</b>	<b>MODIFICA</b>	<b>Definisci</b>	<b>VISUALIZZA</b>	<b>DELETE</b>
<b>comando PCF</b>	<b>Modifica</b>	<b>Crea / Copia</b>	<b>Richiedi</b>	<b>Elimina</b>
AUTHINFO	X	X	X	X
CFSTATO			X	
CFSTRUCT	X	X	X	X
CHANNEL	X	X	X	X
STATO DEL CHF			X	
NAMELIST	X	X	X	X
PROCESS	X	X	X	X
QALIAS	M	M	M	M
QCLUSTER			M	
QLOCALE	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
CODA	P	P	X	P
STATO Q			X	
STGCLASS	X	X	X	X

### Chiave per i simboli della tabella

- M = Solo MQSC
- P = solo PCF
- X = entrambi

Ci sono molti altri comandi MQSC e PCF che consentono di gestire altre risorse IBM MQ ed eseguire altre azioni oltre a quelle riepilogate in [Tabella 25 a pagina 421](#).

Tabella 26 a pagina 422 mostra ogni comando MQSC e dove ogni comando può essere emesso da:

- Dataset di input di inizializzazione CSQINP1
- Dataset di input di inizializzazione CSQINP2
- Console z/OS (o equivalente)
- SYSTEM.COMMAND.INPUT (dalle applicazioni, CSQUTIL o il dataset di input di inizializzazione CSQINPX)

*Tabella 26. Origini da cui eseguire i comandi MQSC*

<b>Comando</b>	<b>CSQINP1</b>	<b>CSQINP2</b>	<b>z/OS console</b>	<b>Coda di immissione comandi e server</b>
MODIFICA AUTHINFO		X	X	X
MODIFICA BUFFERPOOL		X	X	X
MODIFICA CFSTRUCT		X	X	X
MODIFICA CANALE		X	X	X
ELENCO NOMI ALTER		X	X	X
PSID ALTER			X	X
MODIFICA PROCESSO		X	X	X
MODIFICA QALIAS		X	X	X
MODIFICA QLOCAL		X	X	X
Gestore code ALTER		X	X	X
MODIFICA QMODEL		X	X	X
MODIFICA QREMOTE		X	X	X
MODIFICA SICUREZZA	X	X	X	X
ALTER STGCLASS		X	X	X
MODIFICA SUB		X	X	X
ALTER TOPIC		X	X	X
MODIFICA TRACCIA	X	X	X	X
LOG DI ARCHIVIO	X	X	X	X
CFSTRUCT DI BACKUP			X	X
CANCELLA QLOCAL		X	X	X
DEFINE AUTINFO		X	X	X
DEFINISCI BUFFPOOL	X	X		
DEFINE CFSTRUCT		X	X	X
Definire il canale		X	X	X
DEFINE LOG			X	X
DEFINE ELENCO_NOMI		X	X	X
DEFINE PROCESS		X	X	X

Tabella 26. Origini da cui eseguire i comandi MQSC (Continua)

Comando	CSQINP1	CSQINP2	z/OS console	Coda di immissione comandi e server
DEFINE PSID	X		X	X
DEFINE QALIAS		X	X	X
DEFINE QLOCAL		X	X	X
DEFINE QMODEL		X	X	X
DEFINIRE QREMOTE		X	X	X
DEFINE STGCLASS		X	X	X
DEFINE SUB			X	X
DEFINISCI ARGOMENTO		X	X	X
DELETE AUTINFO		X	X	X
ELIMINA BUFFPOOL			X	X
DELETE CFSTRUCT		X	X	X
Elimina canale			X	X
Eliminazione elenco nomi		X	X	X
Eliminazione processo		X	X	X
ELIMINA PSID			X	X
DELETE QALIAS		X	X	X
ELIMINA QLOCALE		X	X	X
DELETE QMODEL		X	X	X
ELIMINA QREMOTE		X	X	X
DELETE STGCLASS		X	X	X
ELIMINA SUB		X	X	X
Elimina argomento		X	X	X
VISUALIZZA ARCHIVIO	X	X	X	X
VISUALIZZA AUTHINFO		X	X	X
VISUALIZZARE CFSTATUS			X	X
VISUALIZZA CFSTRUCT		X	X	X
VISUALIZZA CANALE		X	X	X
VISUALIZZA CHSTATUS			X	X
VISUALIZZA CLUSQMGR			X	X
VISUALIZZA CMDSERV	X	X	X	X
VISUALIZZA CONN		X	X	X
VISUALIZZA CHINIT		X	X	X

Tabella 26. Origini da cui eseguire i comandi MQSC (Continua)

Comando	CSQINP1	CSQINP2	z/OS console	Coda di immissione comandi e server
Visualizza gruppo		X	X	X
VISUALIZZA LOG	X	X	X	X
VISUALIZZA ELENCO NOMI		X	X	X
VISUALIZZA PROCESSO		X	X	X
VISUALIZZA QALIAS		X	X	X
VISUALIZZAZIONE QCLUSTER		X	X	X
VISUALIZZAZIONE QLOCAL		X	X	X
VISUALIZZAZIONE QMGR		X	X	X
VISUALIZZA QMODEL		X	X	X
VISUALIZZAZIONE QREMOTE		X	X	X
VISUALIZZAZIONE QSTATUS		X	X	X
VISUALIZZAZIONE CODA		X	X	X
VISUALIZZAZIONE SICUREZZA			X	X
VISUALIZZA CLASSE DI STG		X	X	X
VISUALIZZA SECONDARIO		X	X	X
VISUALIZZA ARGOMENTO		X	X	X
SISTEMA VIDEO	X	X	X	X
VISUALIZZA THREAD		X	X	X
VISUALIZZA TRACCIA	X	X	X	X
Visualizza utilizzo		X	X	X
SPOSTA QLOCAL		X	X	X
Ping canale			X	X
Recupero BSDS	X	X	X	X
RECUPERO CFSTRUCT			X	X
Aggiornamento cluster		X	X	X
AGGIORNA QMGR		X	X	X
Aggiorna sicurezza		X	X	X
Reimpostazione canale			X	X
Reimposta cluster		X	X	X
QSTATS RESET		X	X	X
Reimpostazione tpipe			X	X
Risoluzione canale			X	X

Tabella 26. Origini da cui eseguire i comandi MQSC (Continua)

Comando	CSQINP1	CSQINP2	z/OS console	Coda di immissione comandi e server
Risoluzione in dubbio		X	X	X
RESUME QMGR			X	X
SICUREZZA RVERIFY		X	X	X
Impostazione archivio	X	X	X	X
Impostazione log	X	X	X	X
Impostazione sistema	X	X	X	X
Avvio canale			X	X
INIZIO INIT		X	X	X
AVVIA CMDSERV	X	X	X	
Avvia listener			X	X
INIZIO GESTORE CODE			X	
Avvio traccia	X	X	X	X
Arresto canale			X	X
ARRESTARE CHINIT			X	X
STOP CMDSERV	X	X	X	
Arresto del listener			X	X
STOP QMGR			X	X
Arresta traccia	X	X	X	X
Gestore code in stato SUSPEND			X	X

Nei comandi MQSC, ogni descrizione di comando identifica le origini da cui è possibile eseguire tale comando.

## Comandi di inizializzazione per IBM MQ for z/OS

I comandi di inizializzazione possono essere utilizzati per controllare l'avvio del gestore code.

I comandi nei dataset di input di inizializzazione vengono elaborati quando IBM MQ viene inizializzato all'avvio del gestore code. È possibile immettere tre tipi di comandi dai dataset di input di inizializzazione:

- Comandi per la definizione di entità IBM MQ che non possono essere definite altrove, ad esempio DEFINE BUFFPOOL.

Questi comandi devono trovarsi nel dataset identificato dal nome DD CSQINP1. Vengono elaborati prima della fase di avvio dell'inizializzazione. Non possono essere emessi tramite la console, le operazioni e i pannelli di controllo o un programma applicativo. Le risposte a questi comandi vengono scritte nel dataset sequenziale a cui si fa riferimento nell'istruzione CSQOUT1 dell'attività avviata.

- Comandi per definire gli oggetti IBM MQ che sono recuperabili dopo il riavvio. Queste definizioni devono essere specificate nel dataset identificato dal nome DD CSQINP2. Vengono memorizzati nella serie di pagine zero. CSQINP2 viene elaborato dopo la fase di riavvio dell'inizializzazione. Le risposte a

questi comandi vengono scritte nel dataset sequenziale a cui si fa riferimento nell'istruzione CSQOUT2 dell'attività avviata.

- Comandi per manipolare gli oggetti IBM MQ . Questi comandi devono essere specificati anche nel dataset identificato dal nome DD CSQINP2. Ad esempio, l'esempio fornito da IBM MQ contiene un comando ALTER QMGR per specificare una coda di messaggi non recapitabili per il sottosistema. La risposta a questi comandi viene scritta nel dataset di output CSQOUT2 .

**Nota:** Se gli oggetti IBM MQ sono definiti in CSQINP2, IBM MQ tenta di ridefinirli ogni volta che il gestore code viene avviato. Se gli oggetti esistono già, il tentativo di definirli ha esito negativo. Se è necessario definire gli oggetti in CSQINP2, è possibile evitare questo problema utilizzando il parametro REPLACE dei comandi DEFINE; tuttavia, questa operazione sovrascrive le eventuali modifiche apportate durante la precedente esecuzione del gestore code.

I membri del dataset di inizializzazione di esempio vengono forniti con IBM MQ for z/OS. Sono descritti in [Definizioni di esempio fornite con IBM MQ](#).

## Comandi di inizializzazione per l'accodamento distribuito

È anche possibile utilizzare il dataset di inizializzazione CSQINP2 per il comando START CHINIT. Se è necessaria una serie di altri comandi per definire il proprio ambiente di accodamento distribuito (ad esempio, l'avvio dei listener), IBM MQ fornisce un terzo dataset di input di inizializzazione, denominato CSQINPX, elaborato come parte della procedura dell'attività avviata dell'iniziatore di canali.

I comandi MQSC contenuti nel dataset vengono eseguiti alla fine dell'inizializzazione dell'iniziatore di canali e l'emissione viene scritta nel dataset specificato dall'istruzione CSQOUTX DD. È possibile utilizzare il dataset di inizializzazione CSQINPX per avviare i listener, ad esempio.

Un membro del dataset di inizializzazione dell'iniziatore di canale di esempio viene fornito con IBM MQ for z/OS. Viene descritto in [Definizioni di esempio fornite con IBM MQ](#).

## Comandi di inizializzazione per la pubblicazione / sottoscrizione

Se hai bisogno di una serie di comandi per definire il tuo ambiente di pubblicazione / sottoscrizione (ad esempio, quando definisci le sottoscrizioni), IBM MQ fornisce un quarto dataset di input di inizializzazione, denominato CSQINPT.

I comandi MQSC contenuti nel dataset vengono eseguiti alla fine dell'inizializzazione di pubblicazione / sottoscrizione e l'emissione viene scritta nel dataset specificato dall'istruzione CSQOUTT DD. È possibile utilizzare il dataset di inizializzazione CSQINPT per definire, ad esempio, le sottoscrizioni.

Un membro del dataset di inizializzazione di pubblicazione / sottoscrizione di esempio viene fornito con IBM MQ for z/OS. Viene descritto in [Definizioni di esempio fornite con IBM MQ](#).

## I programmi di utilità IBM MQ for z/OS

IBM MQ for z/OS fornisce una serie di programmi di utilità che possono essere utilizzati per la gestione del sistema.

IBM MQ for z/OS fornisce una serie di programmi di utilità che consentono di eseguire varie attività di gestione, tra cui:

- Gestire le politiche di protezione dei messaggi.
- Eseguire le attività di backup, ripristino e riorganizzazione.
- Immettere i comandi e le definizioni degli oggetti processo.
- Generare uscite di conversione dati.
- Modificare il dataset di avvio.
- Elencare le informazioni sui log.
- Stampare i log.

- Impostare le tabelle Db2 e altri programmi di utilità Db2 .
- Elaborare i messaggi sulla coda di messaggi non recapitabili.

## **Il programma di utilità della politica di sicurezza del messaggio**

Il programma di utilità della politica di sicurezza dei messaggi (CSQ0UTIL) viene eseguito come programma di utilità autonomo per gestire le politiche di sicurezza dei messaggi. Consultare [Il programma di utilità della politica di sicurezza dei messaggi \(CSQ0UTIL\)](#) per ulteriori informazioni.

## **Programma di utilità CSQUTIL**

Si tratta di un programma di utilità fornito come supporto per le attività di backup, ripristino e riorganizzazione. Per ulteriori informazioni, consultare [Il programma di utilità CSQUTIL](#) .

## **Il programma di utilità di uscita conversione dati**

Programma di utilità IBM MQ for z/OS data conversion exit ( **CSQUCVX** ) viene eseguito come programma di utilità autonomo per creare routine di uscita di conversione dati.

## **Il programma di utilità di inventario del log delle modifiche**

Il programma di utilità IBM MQ for z/OS change log inventory ( **CSQJU003** ) viene eseguito come un programma di utilità autonomo per modificare il dataset bootstrap (BSDS). È possibile utilizzare il programma di utilità per eseguire le seguenti funzioni:

- Aggiungere o eliminare i dataset di log di archivio o attivi.
- Fornire le password per i log di archivio.

## **Il programma di utilità di stampa della mappa del log**

Il programma di utilità di stampa della mappa di log IBM MQ for z/OS ( **CSQJU004** ) viene eseguito come un programma di utilità autonomo per elencare le informazioni riportate di seguito:

- Nome dataset di log e associazione RBA di log per entrambe le copie di tutti i dataset di log attivi e di archivio. Se la registrazione doppia non è attiva, c'è solo una copia dei dataset.
- Dataset di log attivi disponibili per i nuovi dati di log.
- Contenuto della coda dei record di checkpoint nel data set di avvio (BSDS).
- Contenuto del record di cronologia del comando di log di archivio.
- Data / ora del sistema e del programma di utilità.

## **Il programma di utilità di stampa del log**

Il programma di utilità di stampa della registrazione ( **CSQ1LOGP** ) viene eseguito come programma di utilità autonomo. È possibile eseguire il programma di utilità specificando:

- Un dataset bootstrap (BSDS)
- Log attivi (senza BSDS)
- Archivia log (senza BSDS)

## **Il programma di utilità del gruppo di condivisione code**

Il programma di utilità del gruppo di condivisione code ( **CSQ5PQSG** ) viene eseguito come programma di utilità autonomo per configurare le tabelle Db2 ed eseguire altre attività Db2 richieste per i gruppi di condivisione code.

## **Il programma di utilità di preformattazione del log attivo**

Il programma di utilità di preformattazione del log attivo ( **CSQJUFMT** ) formatta i dataset di registrazione attivi prima che vengano utilizzati dal gestore code. Se i dataset di log attivi sono preformattati dal programma di utilità, le prestazioni di scrittura dei log vengono migliorate al primo passaggio del gestore code attraverso i log attivi.

## **Il programma di utilità del gestore code di messaggi non recapitabili**

Il programma di utilità gestore code di messaggi non instradabili ( **CSQUDLQH** ) viene eseguito come programma di utilità autonomo. Controlla i messaggi che si trovano nella coda dei messaggi non instradabili e li elabora in base a una serie di regole fornite al programma di utilità.

## **Il programma di utilità qload**

Da IBM MQ 8.0, il programma di utilità **qload** , fornito in IBM MQ Supportpac MO03, è stato integrato in IBM MQ come programma di utilità **dmpmqmsg** .

Su z/OS, il programma di utilità è disponibile come modulo eseguibile, CSQUDMSG nella libreria SCSQLOAD, con un alias di QLOAD per la compatibilità. JCL di esempio viene fornito anche come membro CSQ4QLOD in SCSQPROC.

## **z/OS programma di utilità CSQUTIL per IBM MQ for z/OS**

Il programma di utilità CSQUTIL viene fornito con IBM MQ for z/OS per consentire di eseguire attività di backup, ripristino e riorganizzazione e per immettere comandi e definizioni di oggetti processo.

Per ulteriori informazioni sul programma di utilità CSQUTIL, consultare [Programma di utilità IBM MQ \(CSQUTIL\)](#). Utilizzando questo programma di utilità, è possibile richiamare le seguenti funzioni:

### **COMANDO**

Per emettere comandi MQSC, registrare definizioni di oggetti e creare file di definizione di canale client.

### **Copia**

Per leggere il contenuto di una coda di messaggi IBM MQ for z/OS denominata o il contenuto di tutte le code di una serie di pagine denominata, inserirli in un file sequenziale e conservare la coda originale.

### **COPYPAGE**

Per copiare intere serie di pagine in serie di pagine più grandi.

### **VUOTO**

Per eliminare il contenuto di una coda messaggi IBM MQ for z/OS denominata o il contenuto di tutte le code di una serie di pagine denominate, conservare le definizioni delle code.

### **FORMATO**

Per formattare le serie di pagine IBM MQ for z/OS .

### **LOAD**

Per ripristinare il contenuto di una coda messaggi IBM MQ for z/OS denominata o il contenuto di tutte le code di una serie di pagine denominate da un file sequenziale creato dalla funzione COPY.

### **PAGINOINFO**

Per estrarre le informazioni sulla serie di pagine da una o più serie di pagine.

## **RESETPAG**

Per copiare intere serie di pagine in altre serie di dati di serie di pagine e ripristinare le informazioni di log nella copia.

## **COPIA**

Copiare il contenuto di una coda in un dataset mentre il gestore code è offline.

## **SDEFS**

Per produrre una serie di comandi di definizione per gli oggetti mentre il gestore code è offline.

## **SLOAD**

Per ripristinare i messaggi dal dataset di destinazione di un'operazione COPY o SCOPY precedente. SLOAD elabora una singola coda.

## **SWITCH**

Per commutare o interrogare la coda di trasmissione associata ai canali mittente del cluster.

## **XPARM**

Per convertire un modulo di caricamento del parametro iniziatore di canale in attributi del gestore code (per scopi di migrazione).

## **z/OS Funzionamento di IBM MQ for z/OS**

Utilizzare queste procedure di base per utilizzare IBM MQ for z/OS.

È inoltre possibile eseguire le operazioni descritte in questa sezione utilizzando IBM MQ Explorer, distribuito con IBM MQ for Windows e IBM MQ per Linux (piattaformex86 e x86-64). Un IBM MQ Explorer autonomo è disponibile per lo scaricamento da Fix Central. Per ulteriori informazioni, consultare [“Amministrazione mediante IBM MQ Explorer”](#) a pagina 114.

## **z/OS Immissione dei comandi del gestore code su z/OS**

È possibile immettere i comandi di controllo IBM MQ da una console z/OS o con il programma di utilità CSQUTIL. I comandi possono utilizzare la stringa del prefisso del comando (CPF) per indicare quale sottosistema IBM MQ elabora il comando.

È possibile controllare la maggior parte dell'ambiente operativo di IBM MQ utilizzando i comandi IBM MQ. IBM MQ for z/OS supporta i tipi MQSC e PCF di tali comandi. Questo argomento descrive come specificare gli attributi utilizzando i comandi MQSC e fa riferimento a tali comandi e attributi utilizzando i relativi nomi di comandi MQSC, piuttosto che i rispettivi nomi PCF. Per informazioni dettagliate sulla sintassi dei comandi MQSC, consultare [Comandi MQSC](#). Per i dettagli sulla sintassi dei comandi PCF, consultare [“Utilizzo di IBM MQ Programmable Command Format”](#) a pagina 25. Se si è un utente autorizzato in modo appropriato, è possibile immettere i comandi IBM MQ da:

- I dataset di input di inizializzazione (descritti in [“Comandi di inizializzazione per IBM MQ for z/OS”](#) a pagina 425).
- Una console z/OS, o equivalente, come SDSF
- La routine del comando get del master z/OS, MGCRE (SVC 34)
- Il programma di utilità IBM MQ, CSQUTIL (descritto nel programma di utilità [IBM MQ](#).)
- Un'applicazione utente, che può essere:
  - Un programma CICS
  - Un programma TSO
  - Un programma batch z/OS
  - Un programma IMS

Consultare [“Scrittura di programmi per amministrare IBM MQ for z/OS”](#) a pagina 450 per informazioni su questo argomento.

La maggior parte delle funzionalità di questi comandi è fornita in modo pratico dalle operazioni e dai pannelli di controllo, accessibili da TSO e ISPF, e descritti in [“Operazioni e pannelli di controllo per IBM MQ for z/OS”](#) a pagina 435.

Per ulteriori informazioni, consultare

- [“Immissione di comandi da una console z/OS o equivalente”](#) a pagina 430
  - [Stringhe del prefisso del comando](#)
  - [Utilizzo della console z/OS per immettere comandi](#)
  - [Risposte comando](#)
- [Immissione di comandi dal programma di utilità CSQUTIL](#)

## Immissione di comandi da una console z/OS o equivalente

Puoi immettere tutti i comandi IBM MQ da una console z/OS o dal suo equivalente. È anche possibile immettere i comandi IBM MQ da qualsiasi punto in cui è possibile immettere i comandi z/OS, ad esempio SDSF o da un programma utilizzando la macro MGCRC.

La quantità massima di dati che è possibile visualizzare come risultato di un comando immesso nella console è 32 KB.

### Nota:

1. Non è possibile immettere comandi IBM MQ utilizzando il formato del comando IMS/SSR da un terminale IMS. Questa funzione non è supportata dall'adattatore IMS.
2. Il campo di input fornito da SDSF potrebbe non essere abbastanza lungo per alcuni comandi, in particolare per i canali.

### Stringhe di prefissi di comandi

Ogni comando IBM MQ deve avere come prefisso una stringa di prefisso del comando (CPF), come mostrato in [Figura 32 a pagina 430](#).

Poiché più di un sottosistema IBM MQ può essere eseguito in z/OS, il CPF viene utilizzato per indicare quale sottosistema IBM MQ elabora il comando. Ad esempio, per avviare il gestore code per un sottosistema denominato CSQ1, dove CPF è '+CSQ1', immettere il comando +CSQ1 START QMGR dalla console dell'operatore. Questo CPF deve essere definito nella tabella dei nomi dei sottosistemi (per il sottosistema CSQ1). Ciò è descritto in [Definizione delle stringhe di prefisso del comando \(CPF\)](#). Negli esempi, la stringa '+CSQ1' viene utilizzato come prefisso del comando.

### Utilizzo della console z/OS per immettere i comandi

È possibile immettere comandi semplici dalla console z/OS, ad esempio il comando DISPLAY in [Figura 32 a pagina 430](#). Tuttavia, per i comandi complessi o per le serie di comandi emessi frequentemente, gli altri metodi di immissione dei comandi sono migliori.

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLLOCAL)
```

*Figura 32. Immissione di un comando DISPLAY dalla console z/OS*

### Risposte comando

Le risposte dirette ai comandi vengono inviate alla console che ha emesso il comando. IBM MQ supporta la funzione *EMCS (Extended Console Support)* disponibile in z/OS, pertanto è possibile utilizzare le console con ID a 4 byte. Inoltre, tutti i comandi tranne START QMGR e STOP QMGR

supportano l'utilizzo dei CART (Command and Response Tokens) quando il comando viene emesso da un programma utilizzando la macro MGCRC.

## Immissione di comandi dal programma di utilità CSQUTIL

È possibile immettere comandi da un dataset sequenziale utilizzando la funzione COMMAND del programma di utilità CSQUTIL. Questo programma di utilità trasferisce i comandi, come messaggi, alla *coda di immissione comandi del sistema* e attende la risposta, che viene stampata insieme ai comandi originali in SYSPRINT. Per i dettagli, consultare [Programma di utilità IBM MQ](#).

### **Avvio e arresto di un gestore code su z/OS**

Utilizzare questo argomento come introduzione all'arresto e all'avvio di un gestore code.

Questa sezione descrive come avviare e arrestare un gestore code. Contiene informazioni sui seguenti argomenti:

- [“Prima di iniziare IBM MQ” a pagina 431](#)
- [“Avvio di un gestore code” a pagina 431](#)
- [“Arresto di un gestore code” a pagina 433](#)

L'avvio e l'arresto di un gestore code è relativamente semplice. Quando un gestore code viene arrestato in condizioni normali, l'ultima azione è quella di eseguire un punto di controllo di terminazione. Questo punto di controllo e i log forniscono al gestore code le informazioni necessarie per il riavvio.

Questa sezione contiene informazioni sui comandi START e STOP e contiene una breve panoramica dell'avvio dopo che si è verificata una chiusura anomala.

### **Prima di iniziare IBM MQ**

Dopo aver installato IBM MQ, viene definito come sottosistema z/OS formale. Questo messaggio viene visualizzato durante qualsiasi IPL (initial program load) di z/OS:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

dove *ssnm* è il nome del sottosistema IBM MQ .

D'ora in poi, è possibile avviare il gestore code per quel sottosistema *da qualsiasi console z/OS che è stata autorizzata a emettere i comandi di controllo del sistema*, ossia un gruppo di comandi SYS z/OS . È necessario emettere il comando START dalla console autorizzata, non è possibile immetterlo tramite JES o TSO.

Se si utilizzano gruppi di condivisione code, è necessario avviare prima RRS e poi Db2, prima di avviare il gestore code.

### **Avvio di un gestore code**

È possibile avviare un gestore code immettendo un comando START QMGR. Tuttavia, non è possibile utilizzare correttamente il comando START a meno che non si disponga dell'autorizzazione appropriata. Consultare [Configurazione della sicurezza su z/OS](#) per informazioni sulla sicurezza IBM MQ . [Figura 33 a](#)

pagina 432 mostra esempi del comando START. (Tenere presente che è necessario aggiungere un prefisso a un comando IBM MQ con una stringa di prefisso del comando (CPF).)

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

Figura 33. Avvio del gestore code da una console z/OS

Consultare [START QMGR](#) per informazioni sulla sintassi del comando START QMGR.

Non è possibile eseguire il gestore code come un lavoro batch o avviarlo utilizzando il comando z/OS START. Questi metodi probabilmente avviano uno spazio di indirizzo per IBM MQ che termina in modo anomalo. Inoltre, non è possibile avviare un gestore code dal programma di utilità CSQUTIL o da un'applicazione utente simile.

Tuttavia, è possibile avviare un gestore code da un programma autorizzato APF inoltrando un comando START QMGR al servizio z/OS MGCRC (SVC 34).

Se si utilizzano gruppi di condivisione code, i sistemi Db2 associati e RRS devono essere attivi quando si avvia il gestore code.

### Opzioni di avvio

Quando si avvia un gestore code, viene caricato un modulo di parametri di sistema. È possibile specificare il nome del modulo dei parametri di sistema in due modi:

- Con il parametro PARM del comando /cpf START QMGR, ad esempio

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- Con un parametro nella procedura di avvio, ad esempio, codificare l'istruzione JCL EXEC come

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARAM(CSQ1ZPRM)'
```

Un modulo di parametri di sistema fornisce le informazioni specificate quando il gestore code è stato personalizzato.

Da IBM MQ 9.1.0, è possibile utilizzare l'opzione **QMGRPROD** per specificare il prodotto rispetto al quale deve essere registrato l'utilizzo del gestore code e l'opzione **AMSPROD** per specificare l'equivalente per AMS, se utilizzato. Consultare il comando MQSC [START QMGR](#) per i dettagli sui valori consentiti.

Di seguito è riportata un'istruzione EXEC JCL di esempio:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

Consultare [z/OS MVS Product Management](#) per ulteriori informazioni sulla registrazione dell'utilizzo del prodotto.

È anche possibile utilizzare l'opzione ENVPARM per sostituire uno o più parametri nella procedura JCL per il gestore code.

Ad esempio, è possibile aggiornare la procedura di avvio del gestore code in modo che il DDname CSQINP2 sia una variabile. Ciò significa che è possibile modificare il DDname CSQINP2 senza modificare la procedura di avvio. Ciò è utile per implementare le modifiche, fornendo i backout per gli operatori e le operazioni dei gestori code.

Si supponga che la procedura di avvio per il gestore code CSQ1 sia simile a [Figura 34 a pagina 433](#).

```

//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
// DD DISP=SHR,DSN=thlqual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqual.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*

```

Figura 34. Procedura di avvio di esempio

Se si avvia il gestore code con il comando:

```
+CSQ1 START QMGR
```

CSQINP2 utilizzato è un membro denominato CSQ1NORM.

Tuttavia, si supponga di mettere in produzione una nuova suite di programmi in modo che al successivo avvio del gestore code CSQ1, le definizioni CSQINP2 vengano prese dal membro CSQ1NEW. A tale scopo, avviare il gestore code con il seguente comando:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

e CSQ1NEW al posto di CSQ1NORM. Nota: z/OS limita le specifiche KEYWORD=value per i parametri simbolici (come in INP2=NEW) a 255 caratteri.

### Avvio dopo una chiusura anomala

IBM MQ rileva automaticamente se il riavvio segue un normale arresto o una chiusura anomala.

L'avvio di un gestore code dopo che è terminato in modo anomalo è diverso dall'avvio dopo l'immissione del comando STOP QMGR. Dopo la STOP QMGR, il sistema termina il suo lavoro in modo ordinato e prende un punto di controllo di terminazione prima dell'arresto. Quando si riavvia il gestore code, esso utilizza le informazioni dal punto di controllo del sistema e dal log di recupero per determinare lo stato del sistema all'arresto.

Tuttavia, se il gestore code termina in modo anomalo, termina senza essere in grado di terminare il lavoro o di eseguire un punto di controllo di terminazione. Quando si riavvia un gestore code dopo una fine anomala, esso aggiorna la conoscenza del relativo stato al termine utilizzando le informazioni nel log e notifica lo stato di varie attività. Normalmente, il processo di riavvio risolve tutti gli stati incongruenti. Tuttavia, in alcuni casi, è necessario adottare misure specifiche per risolvere le incongruenze.

### Messaggi utente all'avvio

Quando un gestore code viene avviato correttamente, il gestore code produce una serie di messaggi di avvio.

### Arresto di un gestore code

Prima di arrestare un gestore code, tutti i messaggi WTOR (write - to - operator - with - reply) correlati a IBM MQ devono ricevere risposte, ad esempio, ottenendo richieste di log. Ogni comando in [Figura 35](#) a [pagina 434](#) termina un gestore code in esecuzione.

```
+CSQ1  STOP QMGR
+CSQ1  STOP QMGR MODE(QUIESCE)
+CSQ1  STOP QMGR MODE(FORCE)
+CSQ1  STOP QMGR MODE(RESTART)
```

*Figura 35. Arresto di un gestore code*

Il comando STOP QMGR assume il valore predefinito STOP QMGR MODE (QUIESCE).

In modalità QUIESCE, IBM MQ non consente la creazione di nuovi thread di connessione, ma consente ai thread esistenti di continuare; termina solo quando tutti i thread sono terminati. Le applicazioni possono richiedere la notifica in caso di sospensione del gestore code. Pertanto, utilizzare la modalità QUIESCE laddove possibile in modo che le applicazioni che hanno richiesto la notifica abbiano l'opportunità di disconnettersi. Per i dettagli, consultare [Cosa accade durante la chiusura](#) .

Se il gestore code non termina in un tempo ragionevole in risposta a un comando STOP QMGR MODE (QUIESCE), utilizzare il comando DISPLAY CONN per determinare se esistono thread di connessione ed eseguire le operazioni necessarie per terminare le applicazioni associate. Se non ci sono sottoprocessi, emettere un comando STOP QMGR MODE (FORCE).

I comandi STOP QMGR MODE (QUIESCE) e STOP QMGR MODE (FORCE) annullano la registrazione IBM MQ da MVS ARM (Automatic Restart Manager), impedendo ad ARM di riavviare automaticamente il gestore code. Il comando STOP QMGR MODE (RESTART) funziona come il comando STOP QMGR MODE (FORCE), ma non annulla la registrazione di IBM MQ da ARM. Ciò significa che il gestore code è idoneo per il riavvio automatico immediato.

Se il sottosistema IBM MQ non è registrato con ARM, il comando STOP QMGR MODE (RESTART) viene rifiutato e il seguente messaggio viene inviato alla console z/OS :

```
CSQY205I ARM element arm-element is not registered
```

Se questo messaggio non viene emesso, il gestore code viene riavviato automaticamente. Per ulteriori informazioni su ARM, consultare [“Utilizzo di z/OS ARM \(Automatic Restart Manager\)”](#) a [pagina 511](#).

**Annulla lo spazio di indirizzo del gestore code solo se STOP QMGR MODE (FORCE) non termina il gestore code.**

Se un gestore code viene arrestato annullando lo spazio di indirizzi o utilizzando il comando STOP QMGR MODE (FORCE), la coerenza viene mantenuta con i sistemi CICS o IMS connessi. La risincronizzazione delle risorse viene avviata quando un gestore code viene riavviato e viene completata quando viene stabilita la connessione al sistema CICS o IMS .

**Nota:** Quando si arresta il gestore code, potrebbe essere visualizzato il messaggio IEF352I . z/OS emette questo messaggio se rileva che se non si contrassegna lo spazio di indirizzo come inutilizzabile, si verificherebbe un'esposizione di integrità. È possibile ignorare questo messaggio.

#### **Messaggi di arresto**

Dopo aver emesso un comando STOP QMGR, si ricevono i messaggi CSQY009I e CSQY002I, ad esempio:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM  
USER(userid), STOP MODE(FORCE)  
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Dove *userid* è l'ID utente che ha emesso il comando STOP QMGR e il parametro MODE dipende da quello specificato nel comando.

Una volta completato correttamente il comando STOP, sulla console z/OS vengono visualizzati i seguenti messaggi:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION  
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

Se si sta utilizzando ARM e non è stato specificato MODE (RESTART), viene visualizzato anche il seguente messaggio:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type  
arm-element-type successful
```

Non è possibile riavviare il gestore code fino a quando non viene visualizzato il seguente messaggio:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```



## Operazioni e pannelli di controllo per IBM MQ for z/OS

È possibile utilizzare le operazioni IBM MQ e i pannelli di controllo per eseguire attività di gestione su oggetti IBM MQ. Utilizzare questo argomento come introduzione ai comandi e ai pannelli di controllo.

Utilizzare questi pannelli per definire, visualizzare, modificare o eliminare gli oggetti IBM MQ. Utilizzare i pannelli per la gestione quotidiana e per apportare piccole modifiche agli oggetti. Se si stanno impostando o modificando molti oggetti, utilizzare la funzione COMMAND del programma di utilità CSQUTIL.

Le operazioni e i pannelli di controllo supportano i controlli per l'iniziatore del canale (ad esempio, per avviare un canale o un listener TCP/IP), per il clustering e per la sicurezza. Consentono inoltre di visualizzare le informazioni sui thread e sull'utilizzo della serie di pagine.

I pannelli funzionano inviando i comandi di tipo MQSC IBM MQ a un gestore code, tramite la coda di input dei comandi di sistema.

### Nota:

1. Le operazioni IBM MQ for z/OS e i pannelli di controllo (CSQOREXX) potrebbero non supportare tutte le nuove funzioni e parametri aggiunti dalla versione 7 in poi. Ad esempio, non ci sono pannelli per la manipolazione diretta di oggetti argomento o sottoscrizioni.

L'uso di uno dei seguenti meccanismi supportati consente di gestire le definizioni di pubblicazione / sottoscrizione e altri controlli di sistema che non sono direttamente disponibili da altri pannelli:

- a. IBM MQ Esplora
- b. z/OS console
- c. Messaggi PCF (Programmable Command Format)
- d. Funzione COMMAND di CSQUTIL

e. Console web IBM MQ

Notare che l'azione generica **Command** nei pannelli CSQOREXX consente di immettere qualsiasi comando MQSC valido, inclusi i comandi relativi a SMDS. È possibile utilizzare tutti i comandi emessi dalla funzione COMMAND di CSQUTIL.

2. Non è possibile immettere i comandi IBM MQ direttamente dalla riga comandi nei pannelli.
3. Per utilizzare le operazioni e i pannelli di controllo, è necessario disporre dell'autorizzazione di sicurezza corretta; ciò è descritto in [ID utente per la sicurezza dei comandi e la sicurezza delle risorse dei comandi](#).
4. Non è possibile fornire un ID utente e una password utilizzando CSQUTIL o i pannelli CSQOREXX. Se invece l'ID utente dispone dell'autorizzazione UPDATE per il profilo BATCH in MQCONN, è possibile ignorare l'impostazione **CHKLOCL** (*REQUIRED*). Per ulteriori informazioni, vedi [Utilizzo di CHKLOCL sulle applicazioni associate localmente](#).

## **Richiamo e regole per le operazioni e i pannelli di controllo**

È possibile controllare IBM MQ e immettere comandi di controllo tramite i pannelli ISPF.

### **Come accedere alle operazioni e ai pannelli di controllo di IBM MQ**

Se il menu delle opzioni primarie ISPF/PDF è stato aggiornato per IBM MQ, è possibile accedere alle operazioni IBM MQ e ai pannelli di controllo da tale menu. Per i dettagli sull'aggiornamento del menu, consultare [Attività 20: Impostazione delle operazioni e dei pannelli di controllo](#).

È possibile accedere alle operazioni e ai pannelli di controllo di IBM MQ dal pannello del programma di elaborazione comandi TSO (in genere l'opzione 6 nel menu delle opzioni primarie ISPF/PDF). Il nome dell'exec eseguito a tale scopo è CSQOREXX. Ha due parametri; `th1qua1` è il qualificatore di alto livello per le librerie IBM MQ da utilizzare e `lang1etter` è la lettera che identifica le librerie di lingua nazionale da utilizzare (ad esempio, E per U.S. Inglese). I parametri possono essere omessi se le librerie IBM MQ sono installate permanentemente nella configurazione ISPF. In alternativa, è possibile immettere CSQOREXX dalla riga comandi TSO.

Questi pannelli sono progettati per essere utilizzati da operatori e amministratori con un minimo di formazione formale. Leggere queste istruzioni con i pannelli in esecuzione e provare le diverse attività suggerite.

**Nota:** Durante l'utilizzo dei pannelli, le code dinamiche temporanee con i nomi nel formato SYSTEM.CSQOREXX.\* vengono creati.

### **Regole per le operazioni e i pannelli di controllo**

Consultare [Regole per la denominazione degli oggetti IBM MQ](#) relative alle regole generali per i nomi e le stringhe di caratteri IBM MQ. Tuttavia, ci sono alcune regole che si applicano solo alle operazioni e ai pannelli di controllo:

- Non racchiudere le stringhe, ad esempio le descrizioni, tra virgolette singole o doppie.
- Se si include un apice o una virgoletta in un campo di testo, non è necessario ripeterlo o aggiungere un carattere di escape. I caratteri vengono salvati esattamente come vengono immessi; ad esempio:

```
This is Maria's queue
```

Il processore del pannello li raddoppia per passare a IBM MQ. Tuttavia, se deve troncare i dati per eseguire questa operazione, lo fa.

- È possibile utilizzare caratteri maiuscoli o minuscoli nella maggior parte dei campi, che vengono piegati in caratteri maiuscoli quando si preme Invio. Le eccezioni sono:

- I nomi delle classi di memoria e i nomi delle strutture CFS (coupling facility structure), che devono iniziare con lettere maiuscole da A a Z e devono essere seguiti da lettere maiuscole da A a Z o da caratteri numerici.
- Alcuni campi non tradotti. Queste includono:
  - ID applicazione
  - Descrizione
  - Dati ambiente
  - Nomi oggetto (ma se si utilizza un nome oggetto in minuscolo, potrebbe non essere possibile immetterlo in una console z/OS )
  - Nome sistema remoto
  - Dati trigger
  - Dati utente
- Nei nomi, gli spazi iniziali e i caratteri di sottolineatura vengono ignorati. Pertanto, non è possibile avere nomi oggetto che iniziano con spazi vuoti o caratteri di sottolineatura.
- I caratteri di sottolineatura vengono utilizzati per visualizzare l'estensione dei campi vuoti. Quando si preme Invio, i caratteri di sottolineatura finali vengono sostituiti da spazi.
- Molti campi di descrizione e testo sono presentati in più parti, ciascuna delle quali è gestita da IBM MQ in modo indipendente. Ciò significa che gli spazi finali vengono conservati e il testo non è contiguo.

### Campi vuoti

Quando si specifica l'azione **Definisci** per un oggetto IBM MQ , ogni campo nel riquadro di definizione contiene un valore. Consultare la guida generale (guida estesa) per i pannelli di visualizzazione per informazioni su dove IBM MQ ottiene i valori. Se si immette un campo con spazi vuoti e gli spazi non sono consentiti, IBM MQ inserisce il valore predefinito di installazione nel campo o richiede di immettere il valore richiesto.

Quando si specifica l'azione **Modifica** per un oggetto IBM MQ , ogni campo sul pannello di modifica contiene il valore corrente per tale campo. Se si immette su un campo con spazi vuoti e questi non sono consentiti, il valore di quel campo non viene modificato.

### **Oggetti e azioni su z/OS**

Le operazioni e i pannelli di controllo offrono diversi tipi di oggetti e una serie di azioni che è possibile eseguire su di essi.

Le azioni sono elencate nel pannello iniziale e consentono di manipolare gli oggetti e visualizzare le informazioni su di essi. Questi oggetti includono tutti gli oggetti IBM MQ , insieme ad altri. Gli oggetti rientrano nelle categorie seguenti.

- Code, processi, oggetti delle informazioni di autenticazione, elenchi nomi, classi di memoria e strutture CF
- Canali
- Oggetti cluster
- Gestore code e sicurezza
- connessioni
- Sistema

Fare riferimento a Azioni per una tabella di riferimento incrociato delle azioni che possono essere eseguite con oggetti IBM MQ .

## **Code, processi, oggetti delle informazioni di autenticazione, elenchi nomi, classi di memoria e strutture CF**

Questi sono gli oggetti IBM MQ di base. Ci possono essere molti di ogni tipo. Possono essere elencati, elencati con filtro, definiti ed eliminati e avere attributi che possono essere visualizzati e modificati, utilizzando le azioni LIST o DISPLAY, LIST con FILTER, DEFINE LIKE, MANAGE e ALTER. (Gli oggetti vengono eliminati utilizzando l'azione MANAGE.)

Questa categoria è composta dai seguenti oggetti:

QLOCALE	Coda locale
QREMOTE	Coda remota
QALIAS	Coda alias per il riferimento indiretto a una coda
QMODEL	Coda modello per la definizione dinamica delle code
CODA	Qualsiasi tipo di coda
STATO Q	Stato di una coda locale
PROCESS	Informazioni su un'applicazione da avviare quando si verifica un evento trigger
AUTHINFO	Informazioni di autenticazione: definizioni richieste per eseguire il controllo CRL (Certificate Revocation List) utilizzando i server LDAP
NAMELIST	Elenco di nomi, come code o cluster
STGCLASS	Classe di memoria
CFSTRUCT	struttura CF (coupling facility)
CFSTATO	Stato di una struttura CF

## **Canali**

I canali vengono utilizzati per l'accodamento distribuito. Ci possono essere molti di ogni tipo, e possono essere elencati, elencati con filtro, definiti, eliminati, visualizzati e modificati. Hanno anche altre funzioni disponibili utilizzando le azioni START, STOP e PERFORM. PERFORM fornisce funzioni di ripristino, ping e risoluzione del canale.

Questa categoria è composta dai seguenti oggetti:

CHANNEL	Qualsiasi tipo di canale
Mittente	Canale di trasmissione
SERVER	Canale server
Ricevente	Canale di ricezione
Richiedente	Canale richiedente
CLUSRCVR	Canale ricevente del cluster
CLUSSDR	Canale mittente del cluster
SVRCONN	Canale di connessione server
CLNTCONN	Canale connessione client
STATO DEL CHF	Stato di una connessione canale

## **Oggetti cluster**

Gli oggetti cluster vengono creati automaticamente per code e canali che appartengono a un cluster. Le definizioni di coda e canale di base possono trovarsi su un altro gestore code. Ci possono essere molti di ciascun tipo e i nomi possono essere duplicati. Possono essere elencati, elencati con filtro e visualizzati. PERFORM, START e STOP sono disponibili anche tramite azioni LIST.

Questa categoria è composta dai seguenti oggetti:

CLUSO	Coda cluster, creata per una coda che appartiene ad un cluster
CLUSCHL	Canale cluster, creato per un canale che appartiene a un cluster
CLUSQMGR	Gestore code cluster, lo stesso di un canale cluster ma identificato dal relativo nome gestore code

I canali cluster e i gestori code cluster hanno le azioni PERFORM, START e STOP, ma solo indirettamente tramite l'azione DISPLAY.

### Gestore code e sicurezza

Il gestore code e gli oggetti di sicurezza hanno una singola istanza. Possono essere elencati e avere attributi che possono essere visualizzati e modificati (utilizzando le azioni LIST o DISPLAY e ALTER) e avere altre funzioni disponibili utilizzando l'azione PERFORM.

Questa categoria è composta dai seguenti oggetti:

Manager	Gestore code: l'azione PERFORM fornisce funzioni di sospensione e ripresa del cluster
Sicurezza	Funzioni di sicurezza: l'azione PERFORM fornisce funzioni di aggiornamento e nuova verifica

### Connessione

Le connessioni possono essere elencate, elencate con filtro e visualizzate.

Questa categoria è costituita solo dall'oggetto connessione, CONNECT.

### Sistema

Una raccolta di altre funzioni. Questa categoria è composta dai seguenti oggetti:

SYSTEM	Funzioni di sistema
CONTROL	Sinonimo di SYSTEM

Le funzioni disponibili sono:

LIST o DISPLAY	Visualizzare il gruppo di condivisione code, l'accodamento distribuito, la serie di pagine o le informazioni sull'utilizzo del dataset.
esecuzione	Aggiorna o reimposta cluster
AVVIA	Avviare l'iniziatore o i listener del canale
ARRESTA	Arresta l'iniziatore o i listener del canale

### Azioni

Le azioni che è possibile eseguire per ciascun tipo di oggetto sono riportate nella tabella seguente:

<i>Tabella 27. Operazioni valide e azioni del pannello di controllo per oggetti IBM MQ.</i>								
Oggetto	Modifica	Definisci come	Gestisci (1)	Elenco o visualizzazione	Elenco con filtro	esecuzione	Avvia	Arresta
AUTHINFO	X	X	X	X	X			
CFSTATO				X				
CFSTRUCT	X	X	X	X	X			

Tabella 27. Operazioni valide e azioni del pannello di controllo per oggetti IBM MQ . (Continua)

Oggetto	Modifica	Definisci come	Gestisci (1)	Elenco o visualizzazione	Elenco con filtro	esecuzione	Avvia	Arresta
CHANNEL	X	X	X	X	X	X	X	X
STATO DEL CHF				X	X			
CLNTCONN	X	X	X	X	X			
CLUSCHL				X	X	X (2)	X (2)	X (2)
CLUSO				X	X			
CLUSQMGR				X	X	X (2)	X (2)	X (2)
CLUSRCVR	X	X	X	X	X	X	X	X
CLUSDR	X	X	X	X	X	X	X	X
CONNECT				X	X			
CONTROL				X		X	X	X
Manager	X			X		X		
NAMELIST	X	X	X	X	X			
PROCESS	X	X	X	X	X			
QALIAS	X	X	X	X	X			
QLOCALE	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QREMOTE	X	X	X	X	X			
STATO Q				X	X			
CODA	X	X	X	X	X			
Ricevente	X	X	X	X	X	X	X	X
Richiedente	X	X	X	X	X	X	X	X
Sicurezza	X			X		X		
Mittente	X	X	X	X	X	X	X	X
SERVER	X	X	X	X	X	X	X	X
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

**Nota:**

1. Fornisce Elimina e altre funzioni
2. Utilizzo dell'azione Elenco o visualizzazione

 **Disposizioni oggetto su z/OS**

È possibile specificare la *disposizione* dell'oggetto che si desidera utilizzare. La disposizione indica dove viene conservato l'oggetto **definizione** e come si comporta l'oggetto.

La disposizione è significativa solo se si sta utilizzando uno dei seguenti tipi di oggetto:

- Code
- canali
- processi
- Elenchi nomi
- Classi di storage
- oggetti delle informazioni di autenticazione

Se si stanno utilizzando altri tipi di oggetto, la disposizione viene ignorata.

I valori consentiti sono:

**Q**

QMGR. Le definizioni degli oggetti si trovano nella serie di pagine del gestore code e sono accessibili solo dal gestore code.

**C**

COPIA. Le definizioni degli oggetti si trovano nella serie di pagine del gestore code e sono accessibili solo dal gestore code. Si tratta di copie locali di oggetti definiti come aventi una disposizione di GROUP.

**P**

Privata. Le definizioni degli oggetti si trovano nella serie di pagine del gestore code e sono accessibili solo dal gestore code. Gli oggetti sono stati definiti come aventi una disposizione QMGR o COPY.

**G**

GRUPPO. Le definizioni degli oggetti si trovano nel repository condiviso e sono accessibili da tutti i gestori code nel gruppo di condivisione code.

**S**

Condiviso. Questa disposizione si riferisce solo alle code locali. Le definizioni di coda sono nel repository condiviso e sono accessibili da tutti i gestori code nel gruppo di condivisione code.

**A**

Tutto. Se il gestore code di azione è il gestore code di destinazione o \*, vengono inclusi gli oggetti di **tutte** le disposizioni; altrimenti, vengono inclusi solo gli oggetti delle disposizioni QMGR e COPY. Questa è l'opzione predefinita.

### **Selezione di un gestore code, dei valori predefiniti e dei livelli utilizzando il pannello di controllo ISPF su z/OS**

È possibile utilizzare l'exec CSQOREXX in ISPF per controllare i gestori code.

Mentre si sta visualizzando il pannello iniziale, non si è connessi ad alcun gestore code. Tuttavia, non appena si preme Invio, si è connessi al gestore code o a un gestore code nel gruppo di condivisione code denominato nel campo **Nome connessione**. È possibile lasciare questo campo vuoto; ciò significa che si sta utilizzando il gestore code predefinito per le applicazioni batch. Ciò è definito in CSQBDEFV (consultare [Attività 19: Impostazione di adattatori batch, TSO e RRS](#) per informazioni su questo argomento).

Utilizzare il campo **Gestore code di destinazione** per specificare il gestore code in cui devono essere eseguite le azioni richieste. Se si lascia vuoto questo campo, per impostazione predefinita viene utilizzato il gestore code specificato nel campo **Nome connessione**. È possibile specificare un gestore code di destinazione che non sia quello a cui ci si connette. In questo caso, di solito si specifica il nome di un oggetto gestore code remoto che fornisce una definizione alias del gestore code (il nome viene utilizzato come *ObjectQMgrName* quando si apre la coda di input del comando). A tale scopo, è necessario che le code e i canali appropriati siano configurati per accedere al gestore code remoto.

Il campo **Gestore code azioni** consente di specificare un gestore code che si trova nello stesso gruppo di condivisione code del gestore code specificato nel campo **Gestore code di destinazione** in modo che sia il gestore code in cui devono essere eseguite le azioni richieste. Se si specifica \* in questo campo, le azioni richieste vengono eseguite su tutti i gestori code nel gruppo di condivisione code. Se si lascia vuoto

questo campo, per impostazione predefinita viene utilizzato il valore specificato nel campo **Gestore code di destinazione** . Il campo **Gestore code azione** corrisponde all'utilizzo del programma di modifica del comando CMDSCOPE descritto in [Comandi MQSC](#).

### Valori predefiniti del gestore code

Se si lasciano vuoti i campi del gestore code o si sceglie di connettersi a un gruppo di condivisione code, viene visualizzata una finestra secondaria quando si preme **Invio**. Questa finestra conferma i nomi dei gestori code che verranno utilizzati. Premere **Invio** per continuare. Quando si ritorna al pannello iniziale dopo aver effettuato alcune richieste, si trovano i campi completati con i nomi effettivi.

### Livelli del gestore code

I pannelli Operazioni e Controllo funzionano in modo soddisfacente solo con i gestori code in esecuzione su z/OS alle IBM WebSphere MQ 710 o successive.

Se queste condizioni non vengono soddisfatte, è probabile che le azioni funzionino solo parzialmente, in modo non corretto o per nulla e che le risposte dal gestore code non vengano riconosciute.

Se il gestore code delle azioni non è uguale o superiore a IBM MQ 8.0.0, alcuni campi non vengono visualizzati e non è possibile immettere alcuni valori. Alcuni oggetti e azioni non sono consentiti. In questi casi, si apre una finestra secondaria che richiede di confermare che si desidera procedere.

## **Utilizzo dei tasti funzione e della riga di comando con i pannelli di controllo ISPF su z/OS**

Per utilizzare i pannelli, è necessario utilizzare i tasti funzione o immettere i comandi equivalenti nell'area comandi del pannello di controllo ISPF.

- [Tasti funzione](#)
  - [Elaborazione delle azioni](#)
  - [“Visualizzazione dei messaggi utente IBM MQ” a pagina 443](#)
  - [Annullamento delle azioni](#)
  - [Come ottenere supporto](#)
- [Utilizzo della riga di comando](#)

### Tasti funzione

I tasti funzionali hanno impostazioni speciali per IBM MQ. Ciò significa che non è possibile utilizzare i valori predefiniti ISPF per i tasti funzione; se in precedenza è stato utilizzato il comando KEYLIST OFF ISPF , è necessario immettere KEYLIST ON nell'area comandi di qualsiasi operazione e pannello di controllo e premere Invio per abilitare le impostazioni IBM MQ .

Queste impostazioni dei tasti funzionali possono essere visualizzate nei pannelli, come mostrato in [Figura 36 a pagina 444](#). Se le impostazioni non vengono visualizzate, immettere PFSHOW nell'area comandi di qualsiasi operazione e pannello di controllo e premere **Invio**. Per rimuovere la visualizzazione delle impostazioni, utilizzare il comando PFSHOW OFF.

Le impostazioni dei tasti funzionali nelle operazioni e nei pannelli di controllo sono conformi agli standard CUA. Sebbene sia possibile modificare l'impostazione della chiave tramite le normali procedure ISPF (come il programma di utilità **KEYLIST** ), non si consiglia di farlo.

**Nota:** L'utilizzo dei comandi **PFSHOW** e **KEYLIST** influisce su qualsiasi altra schermata ISPF logica di cui si dispone e le relative impostazioni rimangono quando si lasciano le operazioni e i pannelli di controllo.

### Elaborazione delle azioni

Premere **Invio** per eseguire l'azione richiesta su un pannello. Le informazioni dal pannello vengono inviate al gestore code per l'elaborazione.

Ogni volta che si preme **Invio** nei pannelli, IBM MQ genera uno o più messaggi operatore. Se l'operazione ha avuto esito positivo, si riceve il messaggio di conferma CSQ9022I, altrimenti si ricevono alcuni messaggi di errore.

### Visualizzazione dei messaggi utente IBM MQ

Premere il tasto funzionale F10 in qualsiasi pannello per visualizzare i messaggi utente IBM MQ .

### Annullamento delle tue azioni

Sul pannello iniziale, sia F3 che F12 escono dalle operazioni e dai pannelli di controllo e ritornano a ISPF. Nessuna informazione viene inviata al gestore code.

Su qualsiasi altro pannello, premere i tasti funzione F3 o F12 per lasciare il pannello corrente **ignorando tutti i dati immessi dall'ultima volta che si è premuto Invio**. Di nuovo, non viene inviata alcuna informazione al gestore code.

- F3 riporta direttamente al pannello iniziale.
- F12 riporta al pannello precedente.

### Come ottenere supporto

A ciascun pannello sono associati dei pannelli di aiuto. I pannelli di aiuto utilizzano i protocolli ISPF:

- Premere il tasto funzionale F1 su qualsiasi pannello per visualizzare la guida generale (guida estesa) relativa all'attività.
- Premere il tasto funzionale F1 con il cursore su qualsiasi campo per visualizzare l'aiuto specifico relativo a tale campo.
- Premere il tasto funzionale F5 da qualsiasi pannello di aiuto del campo per ottenere la guida generale.
- Premere il tasto funzione F3 per tornare al pannello di base, ossia il pannello da cui è stato premuto il tasto funzionale F1.
- Premere il tasto funzionale F6 da qualsiasi pannello di aiuto per ottenere aiuto sui tasti funzionali.

Se le informazioni della guida si inseriscono in una seconda o in pagine successive, viene visualizzato un indicatore **Altro** nella parte superiore destra del pannello. Utilizzare questi tasti funzionali per navigare nelle pagine della guida:

- F11 per andare alla pagina di aiuto successiva (se presente).
- F10 per tornare alla pagina di aiuto precedente (se presente).

### Utilizzo della riga di comando

Non è mai necessario utilizzare la riga comandi per immettere i comandi utilizzati dalle operazioni e dai pannelli di controllo poiché sono disponibili dai tasti funzione. La riga comandi viene fornita per consentire l'immissione di normali comandi ISPF (come **PFSHOW**).

Il comando ISPF PANELID ON visualizza il nome del pannello CSQOREXX corrente.

La riga comandi viene visualizzata inizialmente nella posizione predefinita nella parte inferiore dei pannelli, indipendentemente dalle impostazioni ISPF di cui si dispone. È possibile utilizzare il comando SETTINGS ISPF da qualsiasi operazione e pannello di controllo per modificare la posizione della riga comandi. Le impostazioni vengono memorizzate per le successive sessioni con le operazioni e i pannelli di controllo.

## Utilizzo delle operazioni e dei pannelli di controllo su z/OS

Utilizzare questo argomento per esaminare il pannello di controllo iniziale visualizzato da CSQOREXX

[Figura 36 a pagina 444](#) mostra il pannello visualizzato quando si avvia una sessione di pannello.

```

IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.
Action . . . . . 1      0. List with filter  4. Manage
                          1. List or Display  5. Perform
                          2. Define like     6. Start
                          3. Alter             7. Stop
                          8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A  Q=Qmgr, C=Copy, P=Private, G=Group,
                          S=Shared, A=All

Connect name . . . . . MQ1C - local queue manager or group
Target queue manager . . . MQ1C
                          - connected or remote queue manager for command input
Action queue manager . . . MQ1C - command scope in group
Response wait time . . . . 30 5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2024. All rights reserved.

Command ==>
F1=Help      F2=Split    F3=Exit     F4=Prompt   F9=SwapNext F10=Messages
F12=Cancel

```

Figura 36. Pannello iniziale di controllo e operazioni IBM MQ

Da questo pannello è possibile eseguire azioni quali:

- Scegliere il gestore code locale desiderato e se si desidera immettere i comandi su tale gestore code, su un gestore code remoto o su un altro gestore code nello stesso gruppo di condivisione code del gestore code locale. Sovrascrivere il nome del gestore code se è necessario modificarlo.
- Selezionare l'azione che si desidera eseguire immettendo il numero appropriato nel campo **Azione**.
- Specificare il tipo di oggetto che si desidera gestire. Premere il tasto funzionale F1 per la guida sui tipi di oggetto se non si è certi di cosa si tratta.
- Specificare la disposizione del tipo di oggetto che si desidera gestire.
- Visualizza un elenco di oggetti del tipo specificato. Immettere un asterisco (\*) nel campo **Nome** e premere **Invio** per visualizzare un elenco di oggetti (del tipo specificato) che sono già stati definiti sul gestore code delle azioni. È quindi possibile selezionare uno o più oggetti da gestire in sequenza. Tutte le azioni sono disponibili dall'elenco.

**Nota:** Si consiglia di effettuare le scelte che risultano in un elenco di oggetti visualizzati e quindi di lavorare da tale elenco. Utilizzare l'azione **Visualizza**, poiché è consentita per tutti i tipi di oggetto.

## z/OS Utilizzo di Command Facility su z/OS

Utilizzare l'editor per immettere o modificare i comandi MQSC da passare al gestore code.

Dal pannello principale, CSQOPRIA, selezionare l'opzione **8 Comandoper** avviare la funzione di comando.

Viene visualizzata una sessione di modifica di un file sequenziale, *prefix*.CSQUTIL.COMMANDS, utilizzato come input per la funzione CSQUTIL COMMAND; consultare [Immissione di comandi in IBM MQ](#).

Non è necessario aggiungere ai comandi il prefisso CPF (command prefix string).

È possibile continuare i comandi MQSC sulle righe successive terminando la riga corrente con i caratteri di continuazione + o -. In alternativa, utilizzare la modalità di modifica della riga per fornire i comandi MQSC lunghi o i valori dei valori degli attributi lunghi all'interno del comando.

### modifica riga

Per utilizzare la modifica riga, spostare il cursore sulla riga appropriata nel pannello di modifica e utilizzare **F4** per visualizzare una singola riga in un pannello scorrevole. Una singola riga può essere costituita da un massimo di 32 760 byte di dati.

Per lasciare la modifica della linea:

- **F3 exit** salva le modifiche apportate alla riga ed esce
- **F12 cancel** ritorna al pannello di modifica eliminando le modifiche apportate alla linea.

Per annullare le modifiche apportate alla sessione di modifica, utilizzare **F12 cancel** per terminare la sessione di modifica lasciando invariato il contenuto del file. I comandi non vengono eseguiti.

## Esecuzione di comandi

Una volta terminata l'immissione dei comandi MQSC, terminare la sessione di modifica con **F3 exit** per salvare i contenuti del file e richiamare CSQUTIL per inoltrare i comandi al gestore code. L'output dell'elaborazione del comando è contenuto in *prefix.CSQUTIL.OUTPUT*. Una sessione di modifica si apre automaticamente su questo file in modo da poter visualizzare le risposte. Premere **F3 exit** per uscire da questa sessione e tornare al menu principale.

## Utilizzo degli oggetti IBM MQ su z/OS

Molte delle attività descritte in questa documentazione riguardano la manipolazione degli oggetti IBM MQ. I tipi di oggetto sono gestori code, code, definizioni di processi, elenchi nomi, canali, canali di connessione client, listener, servizi e oggetti delle informazioni di autenticazione.

- [Definizione di oggetti coda semplici](#)
- [Definizione di altri tipi di oggetti](#)
- [Gestione delle definizioni di oggetti](#)
- [Gestione degli elenchi nomi](#)

### Definizione di oggetti coda semplici

Per definire un nuovo oggetto, utilizzare una definizione esistente come base. Puoi farlo in uno dei tre seguenti modi:

- Selezionando un oggetto che è un membro di un elenco visualizzato come risultato delle opzioni selezionate sul pannello iniziale. Si immette quindi il tipo di azione 2 ( **Definisci come** ) nel campo di azione accanto all'oggetto selezionato. Il nuovo oggetto ha gli attributi dell'oggetto selezionato, tranne la disposizione. È quindi possibile modificare gli attributi nel nuovo oggetto in base alle proprie esigenze.
- Sul riquadro iniziale, selezionare il tipo di azione **Definisci come** , immettere il tipo di oggetto che si sta definendo nel campo **Tipo oggetto** e immettere il nome di un oggetto esistente specifico nel campo **Nome** . Il nuovo oggetto ha gli stessi attributi dell'oggetto indicato nel campo **Nome** , ad eccezione della disposizione. È quindi possibile modificare gli attributi nella nuova definizione dell'oggetto in base alle proprie esigenze.
- Selezionando il tipo di azione **Definisci come** , specificando un tipo di oggetto e lasciando vuoto il campo **Nome** . È quindi possibile definire il nuovo oggetto e disporre degli attributi predefiniti definiti per l'installazione. È quindi possibile modificare gli attributi nella nuova definizione dell'oggetto in base alle proprie esigenze.

**Nota:** Non si immette il nome dell'oggetto che si sta definendo nel pannello iniziale, ma nel pannello **Definisci** che viene visualizzato.

Il seguente esempio mostra come definire una coda locale utilizzando una coda esistente come modello.

### Definizione di una coda locale

Per definire un oggetto coda locale dalle operazioni e dai pannelli di controllo, utilizzare una definizione di coda esistente come base per la nuova definizione. Esistono diversi pannelli da completare. Una volta completati tutti i pannelli e si è soddisfatti che gli attributi siano corretti, premere Invio per inviare la propria definizione al gestore code, che a sua volta crea la coda effettiva.

Utilizzare l'azione **Definisci come** sul pannello iniziale o su una voce di oggetto in un elenco visualizzato come risultato delle opzioni selezionate sul pannello iniziale.

Ad esempio, a partire dal pannello iniziale, completare questi campi:

<b>Azione</b>	2 (Definisci come)
<b>Tipo di oggetto</b>	QLOCALE
<b>Nome</b>	QUEUE.YOU.LIKE. Questo è il nome della coda che fornisce gli attributi per la nuova coda.

Premere Invio per visualizzare il pannello **Definisci una coda locale**. Il campo del nome della coda è vuoto in modo da poter fornire il nome per la nuova coda. La descrizione è quella della coda su cui si basa questa definizione. Sovrascrivere questo campo con la propria descrizione per la nuova coda.

I valori negli altri campi sono quelli della coda su cui si basa questa nuova coda, tranne la disposizione. È possibile sovrascrivere questi campi come richiesto. Ad esempio, immettere Y nel campo **Inserisci abilitato** (se non è già Y) se le applicazioni adeguatamente autorizzate possono inserire messaggi in questa coda.

Si ottiene l'aiuto del campo spostando il cursore in un campo e premendo il tasto funzione F1. La guida del campo fornisce informazioni sui valori che possono essere utilizzati per ogni attributo.

Una volta completato il primo pannello, premere il tasto funzione F8 per visualizzare il secondo pannello.

#### **Suggerimenti:**

1. Non premere Invio in questa fase, altrimenti la coda verrà creata prima di poter completare i rimanenti campi. (Se si preme Invio prematuramente, non preoccuparsi; è sempre possibile modificare la definizione in un secondo momento).
2. Non premere i tasti funzionali F3 o F12, altrimenti i dati immessi andranno persi.

Premere ripetutamente il tasto funzionale F8 per visualizzare e completare i pannelli rimanenti, inclusi i pannelli di definizione trigger, di controllo eventi e di notifica backout.

#### **Quando la definizione della coda locale è completa**

Una volta completata la definizione, premere Invio per inviare le informazioni al gestore code per l'elaborazione. Il gestore code crea la coda in base alla definizione fornita. Se non si desidera creare la coda, premere il tasto funzione F3 per uscire e annullare la definizione.

#### **Definizione di altri tipi di oggetti**

Per definire altri tipi di oggetto, utilizzare una definizione esistente come base per la nuova definizione, come descritto in [Definizione di una coda locale](#).

Utilizzare l'azione **Definisci come** sul pannello iniziale o su una voce di oggetto in un elenco visualizzato come risultato delle opzioni selezionate sul pannello iniziale.

Ad esempio, a partire dal pannello iniziale, completare questi campi:

<b>Azione</b>	2 (Definisci come)
<b>Tipo di oggetto</b>	QALIAS, NAMELIST, PROCESS, CHANNEL e altri oggetti risorsa.
<b>Nome</b>	Lasciare vuoto o immettere il nome di un oggetto esistente dello stesso tipo.

Premere Invio per visualizzare i pannelli DEFINE corrispondenti. Completare i campi come richiesto e premere di nuovo Invio per inviare le informazioni al gestore code.

Come la definizione di una coda locale, la definizione di un altro tipo di oggetto generalmente richiede il completamento di diversi pannelli. La definizione di un elenco nomi richiede del lavoro aggiuntivo, come descritto in ["Utilizzo degli elenchi nomi" a pagina 447](#).

#### **Gestione delle definizioni di oggetti**

Una volta definito un oggetto, è possibile specificare un'azione nel campo **Azione** per modificarla, visualizzarla o gestirla.

In ogni caso, è possibile:

- Selezionare l'oggetto che si desidera gestire da un elenco visualizzato come risultato delle opzioni selezionate sul pannello iniziale. Ad esempio, dopo aver immesso 1 nel campo **Azione** per visualizzare gli oggetti, Queue nel campo **Tipo di oggetto** e \* nel campo **Nome**, viene visualizzato un elenco di

tutte le code definite nel sistema. È quindi possibile selezionare da questo elenco la coda con cui è necessario lavorare.

- Iniziare dal pannello iniziale, in cui si specifica l'oggetto che si sta gestendo completando i campi **Tipo di oggetto** e **Nome** .

#### **Modifica di una definizione di oggetto**

Per modificare una definizione di oggetto, specificare l'azione 3 e premere Invio per visualizzare i pannelli ALTER. Questi pannelli sono molto simili ai pannelli DEFINE. È possibile modificare i valori desiderati. Una volta completate le modifiche, premere Invio per inviare le informazioni al gestore code.

#### **Visualizzazione di una definizione di oggetto**

Se si desidera visualizzare i dettagli di un oggetto senza poterli modificare, specificare l'azione 1 e premere Invio per visualizzare i pannelli DISPLAY. Di nuovo, questi pannelli sono simili ai pannelli DEFINE, tranne per il fatto che non è possibile modificare nessuno dei campi. Modificare il nome oggetto per visualizzare i dettagli di un altro oggetto.

#### **Eliminazione di un oggetto**

Per eliminare un oggetto, specificare l'azione 4 (Gestisci) e l'azione **Elimina** è una delle azioni presentate nel menu risultante. Selezionare l'azione **Elimina** .

Ti viene richiesto di confermare la tua richiesta. Se si preme il tasto funzionale F3 o F12, la richiesta viene annullata. Se si preme Invio, la richiesta viene confermata e passata al gestore code. L'oggetto specificato viene quindi cancellato.

**Nota:** Non è possibile eliminare la maggior parte dei tipi di oggetto canale a meno che l'iniziatore di canali non sia avviato.

## **Utilizzo degli elenchi nomi**

Quando si utilizzano gli elenchi nomi, procedere come per gli altri oggetti.

Per le azioni DEFINE LIKE o ALTER, premere il tasto funzione F11 per aggiungere nomi all'elenco o per modificare i nomi nell'elenco. Ciò implica la gestione dell'editor ISPF e tutti i normali comandi di modifica ISPF sono disponibili. Immettere ciascun nome nell'elenco nomi su una riga separata.

Quando si utilizza l'editor ISPF in questo modo, le impostazioni dei tasti funzionali sono le normali impostazioni ISPF e **non** quelle utilizzate da altri pannelli di controllo e operazioni.

Se è necessario specificare nomi minuscoli nell'elenco, specificare CAPS (OFF) sulla riga comandi del pannello dell'editor. Quando si esegue questa operazione, tutti gli elenchi nomi che si modificano in futuro sono in minuscolo fino a quando non si specifica CAPS (ON).

Una volta terminata la modifica dell'elenco nomi, premere il tasto funzione F3 per terminare la sessione di modifica ISPF. Quindi premere Invio per inviare le modifiche al gestore code.

**Attenzione:** Se non si preme Invio in questa fase, ma si preme invece il tasto funzione F3 , si perderanno gli aggiornamenti immessi.

## **Implementazione del sistema utilizzando più code di trasmissione cluster**

Non fa alcuna differenza se il canale viene utilizzato in un singolo cluster o in un cluster sovrapposto. Quando il canale è selezionato e avviato, il canale seleziona la coda di trasmissione in base alle definizioni.

### **Procedura**

- Se si sta utilizzando l'opzione DEFCLXQ, consultare [“Utilizzo della definizione automatica di code e commutazione”](#) a pagina 448.

- Se si utilizza un approccio a fasi, consultare [“Modifica dei canali mittenti del cluster utilizzando un approccio graduale”](#) a pagina 448.

### **z/OS** *Utilizzo della definizione automatica di code e commutazione*

Utilizzare questa opzione se si prevede di utilizzare l'opzione DEFCLXQ. Ci sarà una coda creata per ogni canale e ogni nuovo canale.

#### **Procedura**

1. Rivedere la definizione di SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE e modificare gli attributi se necessario.  
Questa coda è definita nel membro SCSQPROC (csq4insx).
2. Creare il SISTEMA SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE .
3. Applicare le politiche di sicurezza per questa coda modello e il SISTEMA SYSTEM.CLUSTER.TRANSMIT. \* \* .

Per z/OS l'ID utente dell'attività avviata dell'iniziatore di canali richiede:

- Controlla l'accesso a CLASS (MQADMIN) per

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelName
```

- Aggiorna accesso a CLASS (MQQUEUE) per

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelName
```

### **z/OS** *Modifica dei canali mittenti del cluster utilizzando un approccio graduale*

Utilizzare questa opzione se si prevede di utilizzare un approccio a fasi. Questo processo consente di passare ai nuovi canali mittenti del cluster in vari momenti per soddisfare le esigenze della tua azienda.

#### **Prima di iniziare**

- Identificare le proprie applicazioni di business e i canali utilizzati.
- Per le code utilizzate, visualizzare i cluster in cui si trovano.
- Visualizzare i canali per visualizzare i nomi delle connessioni, i nomi dei gestori code remoti e i cluster supportati dal canale.

#### **Informazioni su questa attività**

- Creare una coda di trasmissione. Su z/OS , è possibile considerare quale serie di pagine si utilizza per la coda.
- Impostare la politica di sicurezza per la coda.
- Modificare il monitoraggio della coda in modo da includere questo nome coda.
- Decidere quali canali devono utilizzare questa coda di trasmissione. I canali devono avere un nome simile, quindi caratteri generici ' \* ' nel CLCHNAME identificare il canale.
- Quando si è pronti a utilizzare la nuova funzione, modificare la coda di trasmissione per specificare il nome dei canali per utilizzare questa coda di trasmissione. Ad esempio CLUSTER1.TOPARISo CLUSTER1.\* o \*.TOPARIS
- Avviare i canali

#### **Procedura**

1. Utilizzare il comando DIS CLUSQMGR (xxxx) XMITQ per visualizzare i canali mittenti del cluster definiti nel cluster, dove xxxx è il nome del gestore code remoto.

2. Impostare il profilo di protezione per la coda di trasmissione e fornire l'accesso alla coda all'iniziatore del canale.
3. Definire la coda di trasmissione da utilizzare e specificare USAGE (XMITQ) INDXTYPE (CORRELID) SHARE e CLCHNAME (*valore*)

L'ID utente dell'attività avviata dell'iniziatore del canale ha bisogno del seguente accesso:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel  
update class(MQQUEUE ssid.SYSTEM.CLUSTER.TRANSMIT.channel
```

e l'ID utente che utilizza il comando SWITCH necessita del seguente accesso:

```
alter cl(MQADMIN) ssid.QUEUE.queueName
```

4. Arrestare e riavviare i canali.

La modifica del canale si verifica quando il canale viene avviato utilizzando un comando MQSC oppure si utilizza CSQUTIL. È possibile identificare quali canali devono essere riavviati utilizzando SWITCH CHANNEL (\*) STATUS di CSQUTIL

In caso di problemi all'avvio del canale, arrestare il canale, risolvere i problemi e riavviare il canale.

Si noti che è possibile modificare l'attributo CLCHNAME ogni volta che è necessario.

Il valore di CLCHNAME utilizzato è quello quando il canale viene avviato, quindi è possibile modificare la definizione di CLCHNAME mentre il canale continua a utilizzare le definizioni dal momento in cui è stato avviato. Il canale utilizza la nuova definizione quando viene riavviato.

## Annullamento di una modifica su z/OS

È necessario disporre di un processo per eseguire il backout di una modifica se i risultati non sono quelli previsti.

### Cosa può andare storto?

Se la nuova coda di trasmissione non è quella prevista:

1. Verificare che CLCHNAME sia quello previsto
2. Esaminare il log del lavoro per controllare se il processo di commutazione è terminato. In caso contrario, attendere e controllare la nuova coda di trasmissione del canale in un secondo momento.

Se si utilizzano più code di trasmissione cluster, è importante progettare le definizioni delle code di trasmissione in modo esplicito ed evitare complicate sovrapposizioni di configurazione. In questo modo, è possibile verificare che, in caso di problemi, sia possibile tornare alle code e alla configurazione originali.

Se si riscontrano problemi durante il passaggio all'utilizzo di una coda di trasmissione diversa, è necessario risolvere eventuali problemi prima di poter procedere con la modifica.

Una richiesta di modifica esistente deve essere completata prima di poter effettuare una nuova richiesta di modifica. Ad esempio:

1. Definire una nuova coda di trasmissione con una profondità massima di uno e ci sono 10 messaggi in attesa di essere inviati.
2. Modificare la coda di trasmissione per specificare il nome del canale nel parametro CLCHNAME.
3. Arrestare e riavviare il canale. Il tentativo di spostare i messaggi non riesce e riporta i problemi.
4. Modificare il parametro CLCHNAME sulla coda di trasmissione in modo che sia vuoto.
5. Arrestare e riavviare il canale. Il canale continua a tentare e completare la richiesta originale, quindi il canale continua a utilizzare la nuova coda di trasmissione.
6. È necessario risolvere i problemi e riavviare il canale in modo che lo spostamento dei messaggi venga completato correttamente.

La prossima volta che il canale viene riavviato, acquisirà le eventuali modifiche, quindi se CLCHNAME è stato impostato su spazi vuoti, il canale non utilizzerà la coda di trasmissione specificata.

In questo esempio, la modifica di CLCHNAME sulla coda di trasmissione in spazi vuoti non significa necessariamente che il canale utilizza SYSTEM.CLUSTER.TRANSMIT TRANSMIT, poiché potrebbero esistere altre code di trasmissione il cui parametro CLCHNAME corrisponde al nome del canale. Ad esempio, un nome generico o l'attributo del gestore code DEFCLXQ potrebbero essere impostati sul canale, quindi il canale utilizza una coda dinamica invece di SYSTEM.CLUSTER.TRANSMIT TRANSMIT.

z/OS

## Scrittura di programmi per amministrare IBM MQ for z/OS

È possibile scrivere i programmi di applicazione per gestire un gestore code. Utilizzare questo argomento per comprendere i requisiti per scrivere i propri programmi di gestione.

### Avvio delle informazioni sull'interfaccia di programmazione di utilizzo generale

Questa serie di argomenti contiene suggerimenti e indicazioni per consentire l'immissione di comandi IBM MQ da un programma di applicazione IBM MQ .

**Nota:** In questo argomento, le chiamate MQI vengono descritte utilizzando la notazione in linguaggio C. Per le chiamate tipiche delle chiamate nei linguaggi COBOL, PL/I e assembler, consultare il manuale [Chiamate funzione](#) .

### Capire come funziona tutto

Nella struttura, la procedura per l'emissione di comandi da un programma applicativo è la seguente:

1. Creare un comando IBM MQ in un tipo di messaggio IBM MQ denominato *messaggio di richiesta*. Il comando può essere in formato MQSC o PCF.
2. Invia (utilizzare MQPUT ) questo messaggio ad una coda speciale denominata coda di immissione comandi di sistema. Il processore comandi IBM MQ esegue il comando.
3. Richiama (utilizzare MQGET ) i risultati del comando come *messaggi di risposta* sulla coda di risposta. Questi messaggi contengono i messaggi utente necessari per determinare se il comando è stato eseguito correttamente e, in caso contrario, quali sono stati i risultati.

Quindi è compito del programma applicativo elaborare i risultati.

Questa serie di argomenti contiene:

z/OS

## Preparazione di code per programmi di gestione

I programmi di gestione richiedono un numero di code predefinite per l'immissione del comando di sistema e la ricezione delle risposte.

Questa sezione si applica ai comandi in formato MQSC. Per l'equivalente in PCF, consultare [“Utilizzo di IBM MQ Programmable Command Format”](#) a pagina 25.

Prima di poter emettere qualsiasi chiamata MQPUT o MQGET , è necessario definire e quindi aprire le code che si intende utilizzare.

### Definizione della coda di immissione comandi del sistema

La coda di immissione dei comandi di sistema è una coda locale denominata SYSTEM.COMMAND.INPUT. Il dataset di inizializzazione CSQINP2 fornito, thlqual.SCSQPROC(CSQ4INSG), contiene una definizione predefinita per la coda di input del comando di sistema. Per la compatibilità con IBM MQ su altre piattaforme, un alias di questa coda, denominato SYSTEM.ADMIN.COMMAND.QUEUE . Per ulteriori informazioni, consultare [Definizioni di esempio](#) fornite con IBM MQ .

### Definizione di una coda di risposta

È necessario definire una coda di risposta per ricevere i messaggi di risposta dal processore comandi IBM MQ . Può essere qualsiasi coda con attributi che consentono l'inserimento di messaggi di risposta. Tuttavia, per operazioni normali, specificare questi attributi:

- UTILIZZO (NORMALE)
- NOTRIGGER (a meno che l'applicazione non utilizzi il trigger)

Evitare l'utilizzo di messaggi persistenti per i comandi, ma se si sceglie di farlo, la coda di risposta non deve essere una coda dinamica temporanea.

Il dataset di inizializzazione CSQINP2 fornito, thlqual.SCSQPROC(CSQ4INSG), contiene una definizione per una coda modello denominata SYSTEM.COMMAND.REPLY.MODEL. È possibile utilizzare questo modello per creare una coda di risposta dinamica.

**Nota:** Le risposte generate dal processore comandi possono avere una lunghezza massima di 15 000 byte.

Se si utilizza una coda dinamica permanente come coda di risposta, l'applicazione dovrebbe consentire il completamento di tutte le operazioni PUT e GET prima di tentare di eliminare la coda, altrimenti è possibile restituire MQRC2055 (MQRC\_Q\_NOT\_EMPTY). Se ciò si verifica, ritentare l'eliminazione della coda dopo alcuni secondi.

### Apertura della coda di immissione comandi di sistema

Prima di poter aprire la coda di input del comando di sistema, il programma applicativo deve essere connesso al proprio gestore code. Utilizzare la chiamata MQI MQCONN o MQCONNX per effettuare questa operazione.

Quindi, utilizzare la chiamata MQI MQOPEN per aprire la coda di input del comando di sistema. Per utilizzare questa chiamata:

1. Impostare il parametro **Options** su MQOO\_OUTPUT
2. Impostare i campi del descrittore dell'oggetto MQOD come segue:

**ObjectType**

MQOT\_Q (l'oggetto è una coda)

**ObjectName**

SYSTEM.COMMAND.INPUT

**ObjectQMgrName**

Se si desidera inviare i messaggi di richiesta al gestore code locale, lasciare questo campo vuoto. Ciò significa che i comandi vengono elaborati localmente.

Se si desidera che i comandi IBM MQ vengano elaborati su un gestore code remoto, inserirne qui il nome. È inoltre necessario disporre delle code e dei collegamenti corretti impostati, come descritto in [Accodamento distribuito e cluster](#).

### Apertura di una coda di risposta

Per richiamare le repliche da un comando IBM MQ , è necessario aprire una coda di risposta. Un modo per farlo è specificare la coda modello, SYSTEM.COMMAND.REPLY.MODEL in una chiamata MQOPEN , per creare una coda dinamica permanente come coda di risposta. Per utilizzare questa chiamata:

1. Impostare il parametro **Options** su MQOO\_INPUT\_SHARED
2. Impostare i campi del descrittore dell'oggetto MQOD come segue:

**ObjectType**

MQOT\_Q (l'oggetto è una coda)

**ObjectName**

Il nome del gestore code di replica. Se il nome coda specificato è il nome di un oggetto coda modello, il gestore code crea una coda dinamica.

**ObjectQMGrName**

Per ricevere risposte sul gestore code locale, lasciare questo campo vuoto.

**DynamicQName**

Specificare il nome della coda dinamica da creare.

 **Utilizzo del server dei comandi**

Il server dei comandi è un componente IBM MQ che funziona con il componente processore dei comandi. È possibile inviare messaggi formattati al server dei comandi che interpreta i messaggi, esegue le richieste di gestione e invia le risposte all'applicazione di gestione.

Il server dei comandi legge i messaggi di richiesta dalla coda di immissione dei comandi del sistema, li verifica e li trasmette come comandi validi al processore dei comandi. Il processore dei comandi elabora i comandi e inserisce le risposte come messaggi di risposta nella coda di risposta specificata. Il primo messaggio di risposta contiene il messaggio utente CSQN205I. Per ulteriori informazioni, consultare [“Interpretazione dei messaggi di risposta dal server dei comandi”](#) a pagina 456. Il server dei comandi elabora anche i comandi dell'iniziatore di canali e del gruppo di condivisione code, ovunque vengano emessi.

**Identificazione del gestore code che elabora i comandi**

Il gestore code che elabora i comandi immessi da un programma di gestione è il gestore code proprietario della coda di input dei comandi di sistema in cui viene inserito il messaggio.

**Avvio del server dei comandi**

Di solito, il server dei comandi viene avviato automaticamente all'avvio del gestore code. Diventa disponibile non appena il messaggio CSQ9022I 'START QMGR' NORMAL COMPLETION viene restituito dal comando START QMGR. Il server dei comandi viene arrestato quando tutte le attività connesse sono state disconnesse durante la fase di chiusura del sistema.

È possibile controllare il server dei comandi utilizzando i comandi START CMDSERV e STOP CMDSERV. Per impedire l'avvio automatico del server dei comandi quando IBM MQ viene riavviato, è possibile aggiungere un comando STOP CMDSERV ai dataset di inizializzazione CSQINP1 o CSQINP2. Tuttavia, ciò non è consigliato in quanto impedisce l'elaborazione di qualsiasi comando dell'iniziatore di canali o del gruppo di condivisione code.

Il comando STOP CMDSERV arresta il server dei comandi non appena termina l'elaborazione del messaggio corrente o immediatamente se non viene elaborato alcun messaggio.

Se il server dei comandi è stato arrestato da un comando STOP CMDSERV nel programma, non è possibile elaborare altri comandi dal programma. Per riavviare il server dei comandi, è necessario immettere un comando START CMDSERV dalla console z/OS.

Se si arresta e si riavvia il server dei comandi mentre il gestore code è in esecuzione, tutti i messaggi che si trovano nella coda di input del comando di sistema quando il server dei comandi si arresta vengono elaborati quando il server dei comandi viene riavviato. Tuttavia, se si arresta e si riavvia il gestore code dopo l'arresto del server dei comandi, solo i messaggi persistenti sulla coda di input del comando di sistema vengono elaborati quando il server dei comandi viene riavviato. Tutti i messaggi non persistenti nella coda di input del comando di sistema vengono persi.

**Invio di comandi al server dei comandi**

Per ogni comando, si crea un messaggio contenente il comando, quindi lo si inserisce nella coda di input del comando di sistema.

**Creazione di un messaggio che includa comandi IBM MQ**

È possibile incorporare i comandi IBM MQ in un programma applicativo creando messaggi di richiesta che includono i comandi richiesti. Per ognuno di questi comandi:

1. Creare un buffer contenente una stringa di caratteri che rappresenta il comando.
2. Emettere una richiamo MQPUT specificando il nome del buffer nel parametro **buffer** della chiamata.

Il modo più semplice per farlo in C è definire un buffer utilizzando 'char'. Ad esempio:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

Quando si crea un comando, utilizzare una stringa di caratteri con terminazione null. Non specificare una stringa di prefisso del comando (CPF) all'avvio di un comando definito in questo modo. Ciò significa che non è necessario modificare gli script di comandi se si desidera eseguirli su un altro gestore code. Tuttavia, è necessario considerare che un CPF è incluso in tutti i messaggi di risposta che vengono inseriti nella coda di risposta.

Il server dei comandi chiude tutti i caratteri minuscoli in maiuscolo a meno che non siano racchiusi tra virgolette.

I comandi possono avere una lunghezza massima di 32 762 caratteri.

### **Inserimento di messaggi nella coda di immissione comandi del sistema**

Utilizzare il richiamo MQPUT per inserire messaggi di richiesta contenenti comandi sulla coda di input del comando di sistema. In questa chiamata si specifica il nome della coda di risposta già aperta.

Per utilizzare la chiamata MQPUT :

1. Impostare i seguenti parametri MQPUT :

**Hconn**

La gestione della connessione restituita dalla chiamata MQCONN o MQCONNX .

**Hobj**

L'handle dell'oggetto restituito dalla chiamata MQOPEN per la coda di input del comando di sistema.

**BufferLength**

La lunghezza del comando formattato.

**Buffer**

Il nome del buffer contenente il comando.

2. Impostare questi campi MQMD:

**MsgType**

MQMT\_REQUEST

**Format**

MQFMT\_STRING o MQFMT\_NONE

Se non si sta utilizzando la stessa codepage del gestore code, impostare *CodedCharSetId* come appropriato e impostare MQFMT\_STRING, in modo che il server dei comandi possa convertire il messaggio. Non impostare MQFMT\_ADMIN, poiché in questo modo il comando viene interpretato come PCF.

**ReplyToQ**

Nome della coda di risposta.

**ReplyToQMGr**

Se si desidera che le risposte vengano inviate al gestore code locale, lasciare vuoto questo campo. Se si desidera che i comandi IBM MQ vengano inviati a un gestore code remoto, inserirne qui il nome. È inoltre necessario disporre delle code e dei collegamenti corretti impostati, come descritto in [Accodamento distribuito e cluster](#).

3. Impostare qualsiasi altro campo MQMD, come richiesto. Di solito, è necessario utilizzare messaggi non persistenti per i comandi.

4. Impostare le opzioni *PutMsgOpts* , come richiesto.

Se si specifica MQPMO\_SYNCPOINT (il valore predefinito), è necessario seguire la chiamata MQPUT con una chiamata del punto di sincronizzazione.

#### **Utilizzo di MQPUT1 e della coda di input del comando di sistema**

Se si desidera inserire un solo messaggio nella coda di input del comando di sistema, è possibile utilizzare la chiamata **MQPUT1** . Questa chiamata combina le funzioni di un **MQOPEN**, seguito da un **MQPUT** di un messaggio, seguito da un **MQCLOSE**, tutto in una chiamata. Se si utilizza questa chiamata, modificare i parametri di conseguenza. Consultare [Inserimento di un messaggio su una coda utilizzando la chiamata MQPUT1](#) per i dettagli.

### **Richiamo delle risposte ai comandi**

Il server dei comandi invia una risposta ad una coda di risposta per ogni messaggio di richiesta ricevuto. Qualsiasi applicazione di gestione deve ricevere e gestire i messaggi di risposta.

Quando il processore dei comandi elabora i comandi, i messaggi di risposta vengono inseriti nella coda di risposta specificata nella chiamata MQPUT . Il server dei comandi invia i messaggi di risposta con la stessa persistenza del messaggio di comando ricevuto.

#### **In attesa di una risposta**

Utilizzare il richiamo MQGET per richiamare una risposta dal messaggio di richiesta. Un messaggio di richiesta può produrre diversi messaggi di risposta. Per i dettagli, consultare [“Interpretazione dei messaggi di risposta dal server dei comandi”](#) a pagina 456.

È possibile specificare un intervallo di tempo per cui una chiamata MQGET attende la generazione di un messaggio di risposta. Se non si riceve una risposta, utilizzare l'elenco di controllo che inizia nell'argomento [“Se non si riceve una risposta”](#) a pagina 456.

Per utilizzare la chiamata MQGET :

1. Impostare i seguenti parametri:

##### **Hconn**

La gestione della connessione restituita dalla chiamata MQCONN o MQCONNX .

##### **Hobj**

L'handle dell'oggetto restituito dalla chiamata MQOPEN per la coda di risposta.

##### **Buffer**

Il nome dell'area per ricevere la risposta.

##### **BufferLength**

La lunghezza del buffer per ricevere la risposta. Deve essere un minimo di 80 byte.

2. Per essere certi di ottenere solo le risposte dal comando immesso, è necessario specificare i campi *MsgId* e *CorrelId* appropriati. Questi dipendono dalle opzioni del report, MQMD\_REPORT, specificate nella chiamata MQPUT :

##### **MQRO\_NONE**

Zero binario, '00 ... 00' (24 valori null).

##### **ID\_MSG\_NEW\_MQRO**

Zero binario, '00 ... 00' (24 valori null).

Questo è il valore predefinito se nessuna di queste opzioni è stata specificata.

##### **MQRO\_PASS\_MSG\_ID**

*MsgId* da MQPUT.

##### **MQRO\_NONE**

Il *MsgId* dalla chiamata MQPUT .

### **ID\_COPY\_MQRO\_MSG\_TO\_CORREL\_ID**

Il *MsgId* dalla chiamata MQPUT .

Questo è il valore predefinito se nessuna di queste opzioni è stata specificata.

### **ID\_CORREL\_PASS\_MQRO\_**

Il *CorrelId* dalla chiamata MQPUT .

Per ulteriori dettagli sulle opzioni del report, consultare [Opzioni del report e indicatori del messaggio](#).

3. Impostare i seguenti *GetMsgOpts* campi:

#### **Options**

MQGMO\_WAIT

Se non si utilizza la stessa codepage del gestore code, impostare MQGMO\_CONVERT e impostare *CodedCharSetId* come appropriato in MQMD.

#### **WaitInterval**

Per le risposte dal gestore code locale, provare 5 secondi. Codificato in millisecondi, diventa 5 000. Per le risposte da parte di un gestore code remoto e i comandi di controllo del canale e di stato, provare 30 secondi. Codificato in millisecondi, diventa 30 000.

## **Messaggi eliminati**

Se il server dei comandi rileva che un messaggio di richiesta non è valido, elimina questo messaggio e scrive il messaggio CSQN205I nella coda di risposta denominata. Se non vi è alcuna coda di risposta, il messaggio CSQN205I viene inserito nella coda di messaggi non recapitabili. Il codice di ritorno in questo messaggio indica il motivo per cui il messaggio di richiesta originale non era valido:

- 00D5020F** Non è di tipo MQMT\_REQUEST.
- 00D50210** Ha lunghezza zero.
- 00D50212** È più lungo di 32 762 byte.
- 00D50211** Contiene tutti spazi vuoti.
- 00D5483E** Era necessaria la conversione, ma *Format* non era MQFMT\_STRING.
- Altro** Consultare [Codici del server dei comandi](#)

## **Il descrittore del messaggio di risposta del server dei comandi**

Per qualsiasi messaggio di risposta, sono impostati i seguenti campi del descrittore del messaggio MQMD:

- MsgType* MQMT\_REPLY
- Feedback* MQFB\_NONE
- Encoding* MQEN\_NATIVE
- Priority* Per quanto riguarda MQMD nel messaggio emesso.
- Persistenc* Per quanto riguarda MQMD nel messaggio emesso.  
*e*
- CorrelId* Dipende dalle opzioni del report MQPUT .
- ReplyToQ* Nessuna.

Il server dei comandi imposta il campo *Options* della struttura MQPMO su MQPMO\_NO\_SYNCPOINT. Ciò significa che è possibile richiamare le risposte non appena vengono create, piuttosto che come un gruppo al punto di sincronizzazione successivo.

## Interpretazione dei messaggi di risposta dal server dei comandi

Ogni richiesta elaborata correttamente da IBM MQ produce almeno due messaggi di risposta. Ogni messaggio di risposta contiene un singolo messaggio utente IBM MQ .

La lunghezza di una risposta dipende dal comando immesso. La risposta più lunga che è possibile ottenere proviene da un DISPLAY NAMELIST e può avere una lunghezza massima di 15 000 byte.

Il primo messaggio utente, CSQN205I, contiene sempre:

- Un conteggio delle repliche (in decimali), che è possibile utilizzare come contatore in un loop per ottenere il resto delle risposte. Il conteggio include questo primo messaggio.
- Il codice di ritorno dal preprocessore comandi.
- Un codice di errore, che è il codice di errore del processore comandi.

Questo messaggio non contiene un CPF.

Ad esempio:

```
CSQN205I    COUNT=      4, RETURN=00000000C, REASON=00000008
```

Il campo COUNT è lungo 8 byte ed è giustificato a destra. Inizia sempre alla posizione 18, cioè immediatamente dopo COUNT=. Il campo RETURN è lungo 8 byte in caratteri esadecimali e si trova immediatamente dopo RETURN= alla posizione 35. Il campo REASON ha una lunghezza di 8 byte in caratteri esadecimali ed è immediatamente successivo a REASON= alla posizione 52.

Se il valore RETURN= è 00000000 e il valore REASON= è 00000004, la serie di messaggi di risposta è incompleta. Dopo aver richiamato le repliche indicate dal messaggio CSQN205I , emettere un'ulteriore chiamata MQGET per attendere un'altra serie di risposte. Il primo messaggio nella serie successiva di risposte è di nuovo CSQN205I, che indica quante risposte ci sono e se ce ne sono ancora altre.

Consultare la documentazione di [IBM MQ for z/OS , completamento, e codici di errore](#) per ulteriori dettagli sui singoli messaggi.

Se si utilizza una funzione non in lingua inglese, il testo e il layout delle risposte sono diversi da quelli mostrati qui. Tuttavia, la dimensione e la posizione dei codici di ritorno e conteggio nel messaggio CSQN205I sono uguali.

## Se non si riceve una risposta

È possibile eseguire una serie di operazioni se non si riceve una risposta alla richiesta al server dei comandi.

Se non si riceve una risposta al messaggio di richiesta, utilizzare questo elenco di controllo:

- Il server dei comandi è in esecuzione?
- Il *WaitInterval* è abbastanza lungo?
- Le code di input e di risposta del comando di sistema sono definite correttamente?
- Le chiamate MQOPEN a queste code sono state eseguite correttamente?
- Le code di input e di risposta del comando di sistema sono abilitate per le chiamate MQPUT e MQGET ?
- Si è pensato di aumentare gli attributi MAXDEPTH e MAXMSGL delle code?
- Si stanno utilizzando correttamente i campi *CorrelId* e *MsgId* ?
- Il gestore code è ancora in esecuzione?
- Il comando è stato creato correttamente?
- Tutti i collegamenti remoti sono definiti e funzionano correttamente?
- Le chiamate MQPUT erano definite correttamente?

- La coda di risposta è stata definita come una coda dinamica temporanea invece di una coda dinamica permanente? (Se il messaggio di richiesta è persistente, è necessario utilizzare una coda dinamica permanente per la risposta.)

Quando il server dei comandi genera risposte ma non può scriverle nella coda di risposta specificata, le scrive nella coda di messaggi non recapitabili.

## z/OS Passaggio di comandi utilizzando MGCRC

Con l'autorizzazione appropriata, un programma applicativo può effettuare richieste a più gestori code utilizzando una routine di servizi z/OS .

Se si dispone dell'autorizzazione corretta, è possibile passare i comandi IBM MQ dal programma a più gestori code tramite il servizio MGCRC (SVC 34) z/OS . Il valore di CPF identifica il particolare gestore code a cui è diretto il comando. Per informazioni sui CPF, consultare [User IDs for command security and command resource security](#) e [“Immissione dei comandi del gestore code su z/OS”](#) a pagina 429.

Se si utilizza MGCRC, è possibile utilizzare un comando e un token di risposta (CART) per ottenere le risposte dirette al comando.

## z/OS Esempi di comandi e relative risposte

Utilizzare questo argomento come una serie di esempi di comandi per il server dei comandi e le risposte dal server dei comandi.

Di seguito sono riportati alcuni esempi di comandi che è possibile creare nei messaggi IBM MQ e i messaggi dell'utente che sono le risposte. Se non diversamente specificato, ogni riga della risposta è un messaggio separato.

- [Messaggi da un comando DEFINE](#)
- [Messaggi da un comando DELETE](#)
- [Messaggi dai comandi DISPLAY](#)
- [Messaggi dai comandi con CMDSCOPE](#)
- [Messaggi da comandi che generano comandi con CMDSCOPE](#)

### Messaggi da un comando DEFINE

Il comando riportato di seguito:

```
DEFINE QLOCAL(Q1)
```

produce questi messaggi:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

Questi messaggi di risposta vengono prodotti al normale completamento.

### Messaggi da un comando DELETE

Il comando riportato di seguito:

```
DELETE QLOCAL(Q2)
```

produce questi messaggi:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
CSQM125I +CSQ1 CSQMUQLC QLOCAL (Q2) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D44002'
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

Questi messaggi indicano che una coda locale denominata Q2 non esiste.

### Messaggi dai comandi DISPLAY

I seguenti esempi mostrano le risposte da alcuni comandi DISPLAY.

#### Individuazione del nome della coda di messaggi non instradabili

Se si desidera individuare il nome della coda di messaggi non recapitabili per un gestore code, immettere questo comando da un programma applicativo:

```
DISPLAY QMGR DEADQ
```

Vengono restituiti i seguenti tre messaggi utente, da cui è possibile estrarre il nome richiesto:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE      )
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION
```

#### Messaggi dal comando DISPLAY QUEUE

I seguenti esempi mostrano come i risultati di un comando dipendono dagli attributi specificati in tale comando.

##### Esempio 1

Definire una coda locale utilizzando il comando:

```
DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE
```

Se si immette il seguente comando da un programma applicativo:

```
DISPLAY QUEUE(Q1) SHARE GET DESCR
```

Vengono restituiti questi tre messaggi utente:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1
QLOCAL ) QSGDISP(QMGR  ) TYPE(
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMDMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

**Nota:** Il secondo messaggio, CSQM401I, viene visualizzato qui e occupa quattro righe.

## Esempio 2

Due code hanno nomi che iniziano con la lettera A:

- A1 è una coda locale con il relativo attributo PUT impostato su DISABLED.
- A2 è una coda remota con il suo attributo PUT impostato su ENABLED.

Se si immette il seguente comando da un programma applicativo:

```
DISPLAY QUEUE(A*) PUT
```

Vengono restituiti questi quattro messaggi utente:

```
CSQN205I  COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2                ) TYPE(
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

**Nota:** Il secondo e il terzo messaggio, CSQM401I e CSQM406I, vengono mostrati qui e occupano tre e due righe.

## Messaggi dal comando DISPLAY NAMELIST

Definire un elenco nomi utilizzando il comando:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

Se si immette il seguente comando da un programma applicativo:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

vengono restituiti i seguenti tre messaggi utente:

```
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1              ) QS
GDISP(QMGR  ) NAMCOUNT(    2) NAMES(Q1
,SAMPLE_QUEUE
)
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY NAMELIST' NORMAL COMPLETION
```

**Nota:** Il secondo messaggio, CSQM407I, viene mostrato qui che occupa tre righe.

## Messaggi dai comandi con CMDSCOPE

I seguenti esempi mostrano le risposte dai comandi immessi con l'attributo CMDSCOPE.

### Messaggi dal comando ALTER PROCESS

Il comando riportato di seguito:

```
ALT PRO(V4) CMDSCOPE(*)
```

produce i seguenti messaggi:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'ALT PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ26
CSQM125I !MQ26 CSQMMSGP PROCESS(V4) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E !MQ26 CSQMMSGP FAILURE REASON CODE X'00D44002'
CSQ9023E !MQ26 CSQMMSGP 'ALT PRO' ABNORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP 'ALT PRO' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=0000000C, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

Questi messaggi indicano che il comando è stato immesso sul gestore code MQ25 e inviato a due gestori code (MQ25 e MQ26). Il comando è stato eseguito correttamente su MQ25, ma la definizione del processo non esiste su MQ26, per cui il comando non è riuscito su tale gestore code.

### Messaggi dal comando DISPLAY PROCESS

Il comando riportato di seguito:

```
DIS PRO(V*) CMDSCOPE(*)
```

produce i seguenti messaggi:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ26
CSQM408I !MQ26 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ26 PROCESS(V3) QSGDISP(QMGR)
CSQ9022I !MQ26 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 7, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ25
CSQM408I !MQ25 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ25 PROCESS(V2) QSGDISP(GROUP)
CSQM408I !MQ25 PROCESS(V3) QSGDISP(QMGR)
CSQM408I !MQ25 PROCESS(V4) QSGDISP(QMGR)
CSQ9022I !MQ25 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS PRO' command for CMDSCOPE(*) normal completion
```

Questi messaggi indicano che il comando è stato immesso sul gestore code MQ25 e inviato a due gestori code (MQ25 e MQ26). Vengono visualizzate informazioni su tutti i processi su ciascun gestore code con nomi che iniziano con la lettera V.

### Messaggi dal comando DISPLAY CHSTATUS

Il comando riportato di seguito:

```
DIS CHS(VT) CMDSCOPE(*)
```

produce i seguenti messaggi:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS CHS' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ25
CSQM422I !MQ25 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ25 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ26
CSQM422I !MQ26 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ26 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS CHS' command for CMDSCOPE(*) normal completion
```

Questi messaggi indicano che il comando è stato immesso sul gestore code MQ25 e inviato a due gestori code (MQ25 e MQ26). Vengono visualizzate informazioni sullo stato del canale su ciascun gestore code.

### Messaggi dal comando STOP CHANNEL

Il comando riportato di seguito:

```
STOP CHL(VT) CMDSCOPE(*)
```

produce questi messaggi:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'STOP CHL' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQM134I !MQ25 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS 'STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS 'STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion
```

Questi messaggi indicano che il comando è stato immesso sul gestore code MQ25 e inviato a due gestori code (MQ25 e MQ26). Canale VT arrestato su ciascun gestore code.

### Messaggi dai comandi che generano comandi con CMDSCOPE

Il comando riportato di seguito:

```
DEF PRO(V2) QSGDISP(GROUP)
```

produce questi messaggi:

```

CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQM122I !MQ25 CSQMMSGP ' DEF PRO' COMPLETED FOR QSGDISP(GROUP)
CSQN138I !MQ25 'DEFINE PRO' command generated for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ26
CSQ9022I !MQ26 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DEFINE PRO' command for CMDSCOPE(*) normal completion

```

Questi messaggi indicano che il comando è stato immesso nel gestore code MQ25. Quando l'oggetto è stato creato nel repository condiviso, è stato generato un altro comando che è stato inviato a tutti i gestori code attivi nel gruppo di condivisione code (MQ25 e MQ26).

z/OS

## Gestione delle risorse IBM MQ su z/OS

Utilizzare i link presenti in questo argomento per informazioni su come gestire le risorse utilizzate da IBM MQ for z/OS, ad esempio, la gestione di file di log, dataset, serie di pagine, pool di buffer e strutture CF (Coupling Facility).

Utilizzare i seguenti collegamenti per i dettagli delle diverse attività di gestione che potrebbe essere necessario completare durante l'utilizzo di IBM MQ for z/OS:

- [“Gestione dei log” a pagina 463](#)
- [“Gestione del dataset di bootstrap \(BSDS\)” a pagina 471](#)
- [“Gestione delle serie di pagine” a pagina 479](#)
- [“Come eseguire il backup e il ripristino delle serie di pagine” a pagina 486](#)
- [“Come eseguire il backup e ripristinare le code utilizzando CSQUTIL” a pagina 490](#)
- [“Gestione dei pool di buffer” a pagina 490](#)
- [“Gestione di gruppi di condivisione code e code condivise su z/OS” a pagina 491](#)

### Concetti correlati

Concetti di IBM MQ for z/OS

[“Amministrazione IBM MQ for z/OS” a pagina 417](#)

La gestione dei gestori code e delle risorse associate include le attività che vengono eseguite di frequente per attivare e gestire tali risorse. Scegliere il metodo che si preferisce per gestire i gestori code e le risorse associate.

[“Immissione di comandi in IBM MQ for z/OS” a pagina 418](#)

È possibile utilizzare i comandi script IBM MQ (MQSC) in modalità batch o interattiva per controllare un gestore code.

[“Ripristino e riavvio su z/OS” a pagina 501](#)

Utilizzare questo argomento per comprendere i meccanismi di ripristino e riavvio utilizzati da IBM MQ.

### Attività correlate

[Pianificazione dell'ambiente IBM MQ su z/OS](#)

[Configurazione dei gestori code su z/OS](#)

[Utilizzo dei programmi di utilità IBM MQ for z/OS](#)

### Riferimenti correlati

[“Origini da cui è possibile emettere comandi MQSC su z/OS” a pagina 20](#)

I comandi MQSC possono essere emessi da varie origini, a seconda del comando.

[“I programmi di utilità IBM MQ for z/OS” a pagina 426](#)

IBM MQ for z/OS fornisce una serie di programmi di utilità che possono essere utilizzati per la gestione del sistema.

## **Gestione dei log**

Utilizzare questo argomento per comprendere come gestire i file di log IBM MQ , incluso il processo di archiviazione dei log, utilizzando la compressione dei record di log, il recupero dei record di log e la stampa dei record di log.

Questo argomento descrive le attività coinvolte nella gestione dei log IBM MQ . Contiene le seguenti sezioni:

## **Archiviazione dei log con il comando ARCHIVE LOG**

Un operatore autorizzato può archiviare i dataset di log attivi IBM MQ correnti quando richiesto utilizzando il comando **ARCHIVE LOG** .

Quando si immette il comando **ARCHIVE LOG** , IBM MQ tronca i dataset di log attivi correnti, quindi esegue un processo di offload asincrona e aggiorna il BSDS con un record del processo di offload.

Il comando **ARCHIVE LOG** ha una opzione **MODE(QUIESCE)** . Con questa opzione, i lavori IBM MQ e gli utenti vengono disattivati dopo un punto di commit e il punto di congruenza risultante viene acquisito nel log attivo corrente prima che venga scaricato.

Considerare l'utilizzo dell'opzione **MODE(QUIESCE)** quando si pianifica una strategia di backup per il ripristino fuori sede. Crea un punto di congruenza a livello di sistema, che riduce il numero di incongruenze dei dati quando il log di archivio viene utilizzato con la copia della serie di pagine di backup più recente durante il ripristino. Ad esempio:

```
ARCHIVE LOG MODE(QUIESCE)
```

Se si immette il comando **ARCHIVE LOG** senza specificare un parametro **TIME** , il periodo di tempo di sospensione assume il valore predefinito del parametro **QUIESCE** della macro CSQ6ARVP . Se il tempo richiesto per il completamento di **ARCHIVE LOG MODE(QUIESCE)** è inferiore al tempo specificato, il comando viene completato correttamente; altrimenti, il comando ha esito negativo quando scade il periodo di tempo. È possibile specificare esplicitamente il periodo di tempo utilizzando l'opzione **TIME** , ad esempio:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

Questo comando specifica un periodo di inattività fino a 60 secondi prima che si verifichi l'elaborazione **ARCHIVE LOG** .

**Attenzione:** L'uso dell'opzione **TIME** quando il tempo è critico può interrompere in modo significativo la disponibilità IBM MQ per tutti i lavori e gli utenti che utilizzano risorse IBM MQ .

Per impostazione predefinita, il comando viene elaborato in modo asincrono dal momento in cui si inoltra il comando. (Per elaborare il comando in modo sincrono con gli altri comandi IBM MQ utilizzare l'opzione **WAIT (YES)** con **QUIESCE**, ma tenere presente che la console z/OS è bloccata dall'input del comando IBM MQ per l'intero periodo **QUIESCE** .)

Durante il periodo di inattività:

- I lavori e gli utenti sul gestore code possono eseguire l'elaborazione del commit, ma vengono sospesi se tentano di aggiornare qualsiasi risorsa IBM MQ dopo il commit.
- I lavori e gli utenti che leggono solo i dati possono essere influenzati, poiché potrebbero essere in attesa di blocchi congelati da lavori o utenti che sono stati sospesi.
- Le nuove attività possono iniziare, ma non possono aggiornare i dati.

L'output del comando **DISPLAY LOG** utilizza il messaggio CSQV400I per indicare che è attivo un quiesce.

```

CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter  Initial value      SET value
-----
INBUFF     60
OUTBUFF    400
MAXRTU     2
MAXARCH    2
TWOACTV    YES
TWOARCH    YES
TWOBSDS    YES
OFFLOAD    YES
MAXCNOFF   0
WRTHRS    20
DEALLCT    0
COMPLLOG   NONE
ZHYWRITE   NO
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full zHyperWrite Encrypted DSName
  1     68 NO          NO          VICY.CSQ1.LOGCOPY1.DS01
  2     68 NO          NO          VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2019-08-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION

```

Ad esempio, per Long Term Support e Continuous Delivery prima di IBM MQ 9.1.2:

```

CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter  Initial value      SET value
-----
INBUFF     60
OUTBUFF    400
MAXRTU     2
MAXARCH    2
TWOACTV    YES
TWOARCH    YES
TWOBSDS    YES
OFFLOAD    YES
MAXCNOFF   0
WRTHRS    20
DEALLCT    0
COMPLLOG   NONE
ZHYWRITE   NO                      YES
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full PPRC DSName
  1     68 NO          VICY.CSQ1.LOGCOPY1.DS01
  2     68 NO          VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2014-04-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION

```

Quando tutti gli aggiornamenti vengono sospesi, il record della cronologia di sospensione in BSDS viene aggiornato con la data e l'ora in cui i dataset di log attivi sono stati troncati e con l'ultimo RBA scritto nei dataset di log attivi correnti. IBM MQ tronca i dataset di log attivi correnti, passa ai successivi dataset di log attivi disponibili ed emette il messaggio CSQJ311I che indica che il processo di offload è stato avviato.

Se non è possibile sospendere gli aggiornamenti prima della scadenza del periodo di sospensione, IBM MQ emette il messaggio CSQJ317I e l'elaborazione **ARCHIVE LOG** termina. I dataset di log attivi correnti non vengono troncati, né commutati ai successivi dataset di log disponibili e il processo di offload non viene avviato.

Se il quiesce ha avuto esito positivo o negativo, tutti gli utenti e i lavori sospesi vengono quindi ripristinati e IBM MQ emette il messaggio CSQJ312I, che indica che il quiesce è terminato e che l'attività di aggiornamento è ripresa.

Se **ARCHIVE LOG** viene immesso quando il log attivo corrente è l'ultimo dataset di log attivo disponibile, il comando non viene elaborato e IBM MQ emette il seguente messaggio:

```
CSQJ319I - csect - name CURRENT ACTIVE LOG DATA SET IS LAST
DATASET LOG ATTIVO DISPONIBILE. ELABORAZIONE LOG ARCHIVIO
VERRÀ TERMINATO
```

Se **ARCHIVE LOG** viene immesso quando un altro comando **ARCHIVE LOG** è già in corso, il nuovo comando non viene elaborato e IBM MQ emette il seguente messaggio:

```
CSQJ318I - COMANDO DI ARCHIVIAZIONE LOG GIÀ IN CORSO
```

Per informazioni sui messaggi emessi durante l'archiviazione, consultare [Messaggi per IBM MQ for z/OS](#).

## Riavvio del processo di archiviazione log dopo un errore

Se si verifica un problema durante il processo di archiviazione del log (ad esempio, un problema con l'assegnazione o il montaggio del nastro), l'archiviazione del log attivo potrebbe essere sospesa. Puoi annullare il processo di archiviazione e riavviarlo utilizzando il seguente comando:

```
ARCHIVE LOG CANCEL OFFLOAD
```

Questo comando annulla qualsiasi elaborazione di offload attualmente in corso e riavvia il processo di archivio. Inizia con il dataset di log meno recente che non è stato archiviato e procede attraverso tutti i dataset di log attivi che devono essere scaricati. Tutte le operazioni di archiviazione log che sono state sospese vengono riavviate.

Utilizzare questo comando solo se si è certi che l'attività di archiviazione log corrente non sia più in funzione o se si desidera riavviare un tentativo precedente non riuscito. Ciò è dovuto al fatto che il comando potrebbe causare una chiusura anomala dell'attività di scaricamento, che potrebbe causare un dump.

## **Controllo dell'archiviazione e della registrazione log**

È possibile controllare la compressione, la stampa, l'archivio, il recupero e la registrazione utilizzando le macro CSQ6LOGP, CSQ6ARVPe CSQ6SYSP . Nota: le modifiche solo agli oggetti privati vengono registrate nei log IBM MQ. Vengono registrate anche le modifiche agli oggetti GROUP (come i canali in ingresso condivisi), perché le definizioni sono propagate intorno al gruppo e conservate localmente.

Molti aspetti dell'archiviazione e della registrazione sono controllati dai parametri impostati utilizzando la macro CSQ6LOGP, CSQ6ARVP e CSQ6SYSP del modulo dei parametri di sistema quando il gestore code è personalizzato. Consultare [Modulo dei parametri di sistema](#) per i dettagli di queste macro.

Alcuni di questi parametri possono essere modificati mentre un gestore code è in esecuzione utilizzando i comandi MQSC SET LOG, SET SYSTEM e SET ARCHIVE di IBM MQ . Sono mostrati in [Tabella 28 a pagina 465](#):

comando set	Parametri
LOG	WRTHRSH, MAXARCH, DEALLCT, MAXRTU, COMPLOG
ARCHIVIO	Tutto
SYSTEM	LOGLOAD

È possibile visualizzare le impostazioni di tutti i parametri utilizzando i comandi MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) e [DISPLAY SYSTEM](#) . Questi comandi mostrano anche le informazioni di stato relative all'archiviazione e alla registrazione.

## Controllo della compressione dei log

È possibile abilitare e disabilitare la compressione dei record di log utilizzando

- I comandi SET e DISPLAY LOG in MQSC; consultare [Comandi MQSC](#)
- Richiamo dell'interfaccia PCF. Vedere "[Introduzione a IBM MQ Programmable Command Formats](#)" a pagina 24
- Utilizzo della macro CSQ6LOGP nel modulo dei parametri di sistema; consultare [Utilizzo di CSQ6LOGP](#)

## Stampa dei record di log

È possibile estrarre e stampare i record di log utilizzando l'utilità CSQ1LOGP . Per istruzioni, consultare [Il programma di utilità di stampa log](#).

## Ripristino dei log

Di solito, non è necessario eseguire il backup e il ripristino dei log IBM MQ , soprattutto se si utilizza la registrazione doppia. Tuttavia, in rare circostanze, come un errore I/O su un log, potrebbe essere necessario ripristinare i log. Utilizzare Access Method Services per eliminare e ridefinire il dataset, quindi copiare il log doppio corrispondente in esso.

### **Eliminazione dei dataset di log di archivio**

È possibile eliminare i dataset di log di archivio e scegliere di eliminare i log automaticamente o manualmente.

È necessario conservare dati di log sufficienti per poter eseguire il ripristino dell'unità di lavoro, il ripristino del supporto della serie di pagine se si perde una serie di pagine o il ripristino del supporto della struttura CF se si perde una struttura CF. Non eliminare i dataset di log di archivio che potrebbero essere richiesti per il recupero; se si eliminano questi dataset di log di archivio, potrebbe non essere possibile eseguire le operazioni di ripristino richieste.

Se è stato confermato che i dataset di log di archivio possono essere eliminati, è possibile eseguire questa operazione in uno dei modi riportati di seguito:

- [Eliminazione automatica dataset di log di archivio](#)
- [Eliminazione manuale dei dataset di log di archiviazione](#)

## Eliminazione automatica del dataset di log di archivio

È possibile utilizzare un DASD o un sistema di gestione nastro per eliminare automaticamente i dataset del log di archivio. Il periodo di conservazione per i dataset di log di archiviazione IBM MQ è specificato dal campo del periodo di conservazione ARCRETN nella macro di installazione CSQ6ARVP (per ulteriori informazioni, consultare [Utilizzo di CSQ6ARVP](#)).

Il valore predefinito per il periodo di conservazione specifica che i log di archiviazione devono essere conservati per 9999 giorni (il massimo).

**Importante:** È possibile cambiare il periodo di conservazione, ma è necessario assicurarsi di poter gestire il numero di cicli di backup pianificati.

.

IBM MQ utilizza il valore del periodo di conservazione come valore per il parametro JCL RETPD quando vengono creati i dataset del log di archiviazione.

Il periodo di conservazione impostato da SMS (storage management subsystem) MVS/DFP può essere sovrascritto da questo parametro IBM MQ . In genere, il periodo di conservazione è impostato sul valore

più piccolo specificato da IBM MQ o SMS. L'amministratore dell'archiviazione e l'amministratore IBM MQ devono concordare un valore del periodo di conservazione appropriato per IBM MQ.

**Nota:** IBM MQ non dispone di un metodo automatizzato per eliminare le informazioni sui dataset di log di archiviazione da BSDS, poiché alcuni sistemi di gestione nastri forniscono sostituzioni manuali esterne dei periodi di conservazione. Pertanto, le informazioni su un dataset di log di archiviazione possono essere ancora presenti in BSDS molto tempo dopo la scadenza del periodo di conservazione del dataset e dopo che il dataset è stato graffiato dal sistema di gestione nastri. Al contrario, il numero massimo di dataset di log di archivio potrebbe essere stato superato e i dati da BSDS potrebbero essere stati eliminati prima che il dataset abbia raggiunto la data di scadenza.

Se i dataset di log di archivio vengono eliminati automaticamente, tenere presente che l'operazione non aggiorna l'elenco di log di archivio in BSDS. È possibile aggiornare il BSDS con il programma di utilità di inventario del log di modifica, come descritto in [“Modifica di BSDS”](#) a pagina 473. L'aggiornamento non è essenziale. La registrazione dei vecchi log di archivio spreca spazio in BSDS, ma non fa altri danni.

## Eliminazione manuale dei dataset di log di archiviazione

È necessario conservare tutti i record di log fino al livello più basso identificato nei messaggi CSQI024I e CSQI025I. Questo RBA si ottiene utilizzando il comando DISPLAY USAGE immesso durante la creazione di un punto di ripristino utilizzando [Metodo 1: backup completo](#).

Leggi [Creazione di un punto di recupero per le risorse non condivise prima di eliminare i log](#).

### Individuare ed eliminare i dataset di log di archiviazione

Dopo aver stabilito l'RBA di log minimo richiesto per il ripristino, è possibile trovare i dataset di log di archiviazione che contengono solo record di log precedenti eseguendo la seguente procedura:

1. Utilizzare il programma di utilità di stampa della mappa del log per stampare il contenuto di BSDS. Per un esempio dell'output, vedere [Programma di utilità di stampa della mappa del log](#).
2. Individuare le sezioni dell'output intitolato ARCHIVE LOG COPY n DATA SETS. Se si utilizza la registrazione doppia, esistono due sezioni. Le colonne etichettate STARTRBA e ENDRBA mostrano l'intervallo di RBA contenuto in ciascun volume. Individuare i volumi con intervalli che includono l'RBA minimo rilevato con messaggi CSQI024I e CSQI025I. Questi sono i primi volumi da conservare. Se si utilizza la registrazione doppia, esistono due volumi di questo tipo.

Se nessun volume ha un intervallo appropriato, si applica uno dei casi seguenti:

- L'RBA minimo non è stato ancora archiviato ed è possibile eliminare tutti i volumi di log di archivio.
- L'elenco dei volumi di log di archiviazione in BSDS racchiuso quando il numero di volumi ha superato il numero consentito dal parametro MAXARCH della macro CSQ6LOGP. Se il BSDS non registra un volume di log di archivio, tale volume non può essere utilizzato per il ripristino. Pertanto, considerare l'aggiunta di informazioni sui volumi esistenti a BSDS. Per le istruzioni, fare riferimento a [“Modifiche per i log di archiviazione”](#) a pagina 475.

Considerare anche l'aumento del valore di MAXARCH. Per informazioni, consultare [Utilizzo di CSQ6LOGP](#).

3. Eliminare qualsiasi volume o dataset di log di archivio con un valore ENDRBA inferiore al valore STARTRBA del primo volume che si desidera conservare. Se si utilizza la registrazione doppia, eliminare entrambe le copie.

Poiché le voci BSDS si riavvolgono, le prime voci nella sezione del log di archiviazione BSDS potrebbero essere più recenti delle voci alla fine. Guarda la combinazione di data e ora e confronta le loro età. Non presumere che sia possibile eliminare tutte le voci prima della voce per il log di archiviazione contenente il LOGRBA minimo.

Eliminare i dataset. Se gli archivi sono su nastro, cancellare i nastri. Se si trovano su DASD, eseguire un programma di utilità z/OS per eliminare ogni dataset. Quindi, se si desidera che BSDS elenchi solo i volumi di archivio esistenti, utilizzare il programma di utilità di inventario del log

di modifica (CSQJU003) per eliminare le voci per i volumi eliminati. Per un esempio, consultare [“Modifiche per i log di archiviazione” a pagina 475.](#)

## **L'effetto dello shunting del log**

Le transazioni di lunga durata possono causare record di unità di lavoro che si estendono ai dataset di log. IBM MQ gestisce questo scenario utilizzando lo smistamento log, una tecnica che sposta i record di log per ottimizzare la quantità di dati di log conservati e il tempo di riavvio del gestore code.

Quando un'unità di lavoro è considerata lunga, una rappresentazione di ogni record di log viene scritta più in basso nel log. Questo è noto come *shunting di log*. È descritto in modo più completo in [File di log](#).

Il gestore code utilizza questi record di log deviati invece degli originali dopo un malfunzionamento, per garantire l'integrità dell'unità di lavoro. Ci sono due vantaggi:

- la quantità di dati di registro che devono essere conservati per il coordinamento dell'unità di lavoro è ridotta
- un numero inferiore di dati di log deve essere attraversato al momento del riavvio del gestore code, in modo che il gestore code venga riavviato più rapidamente

I record di log deviati non contengono informazioni sufficienti per le operazioni di ripristino dei supporti.

I dati contenuti nel log vengono utilizzati per due scopi distinti: recupero dei supporti e coordinamento dell'unità di lavoro. Se si verifica un errore del supporto che influisce su una struttura CF o su una serie di pagine, il gestore code può ripristinare il supporto al punto di errore ripristinando una copia precedente e aggiornandolo utilizzando i dati contenuti nel log. L'attività persistente eseguita in un'unità di lavoro viene registrata nel log in modo che, in caso di errore, possa essere ripristinata o i blocchi possano essere ripristinati sulle risorse modificate. La quantità di dati di log che è necessario conservare per abilitare il recupero del gestore code è influenzata da questi due elementi.

Per il ripristino dei supporti, è necessario conservare dati di log sufficienti per poter eseguire il ripristino dei supporti almeno dalla copia dei supporti più recente e per poter eseguire il backout. (Il tuo sito può prevedere la possibilità di recuperare da backup più vecchi.) Per l'integrità dell'unità di lavoro, è necessario conservare i dati di log per le unità di lavoro più vecchie o in dubbio.

Per facilitare la gestione del sistema, il gestore code rileva le vecchie unità di lavoro in ciascun archivio di log e le riporta nei messaggi CSQJ160 e CSQJ161. Un'attività interna legge le informazioni del registro delle unità di lavoro per queste vecchie unità di lavoro e le riscrive in un formato più succinto nella posizione corrente nel registro. Il messaggio CSQR026 indica quando ciò si è verificato. Il comando MQSC DISPLAY USAGE TYPE (DATASET) consente anche di gestire la conservazione dei dati di log. Il comando riporta le seguenti tre informazioni di ripristino:

1. La quantità di log che deve essere conservata per il ripristino dell'unità di lavoro.
2. La quantità di log che deve essere conservata per il ripristino dei supporti delle serie di pagine.
3. Per un gestore code in un gruppo di condivisione code, quanta parte del log deve essere conservata per il recupero del supporto delle strutture CF.

Per ciascuna di queste informazioni, viene effettuato un tentativo di mappare i dati di log meno recenti richiesti in un dataset. Poiché le nuove unità di lavoro vengono avviate e arrestate, (1) si prevede di passare a una posizione più recente nel log. Se non si sta spostando, i messaggi UOW di lunga durata avvisano l'utente che si è verificato un problema. (2) si riferisce al ripristino del supporto della serie di pagine se il gestore code deve essere arrestato ora e riavviato. Non è noto quando è stato eseguito l'ultimo backup delle serie di pagine o quale backup potrebbe essere necessario utilizzare se si è verificato un errore della serie di pagine. Normalmente si sposta in una posizione più recente nel log durante l'elaborazione del punto di controllo quando le modifiche contenute nei pool di buffer vengono scritte nelle serie di pagine. In (3), il gestore code è a conoscenza dei backup delle strutture CF eseguiti su questo gestore code o su altri gestori code nel gruppo di condivisione code. Tuttavia, il ripristino della struttura CF richiede un'integrazione dei dati di log da tutti i gestori code nel gruppo di condivisione code che hanno interagito con la struttura CF dall'ultimo backup. Ciò significa che i dati di log sono identificati da un numero di sequenza del record di log (o LRSN), che è basato sulla data / ora e quindi applicabile nell'intero gruppo di condivisione code piuttosto che da un RBA che sarebbe diverso su gestori code differenti nel gruppo di condivisione code. Normalmente si sposta in una posizione più recente nel log

quando i comandi BACKUP CFSTRUCT vengono eseguiti su questo o su altri gestori code nel gruppo di condivisione code.

## **Reimpostazione del log del gestore code**

Utilizzare questo argomento per comprendere come reimpostare il log del gestore code.

È necessario non consentire all'RBA del log del gestore code di eseguire il wrap dalla fine dell'intervallo di RBA del log a 0, poiché ciò porta a un'interruzione del gestore code e tutti i dati persistenti diventeranno irrecuperabili. La fine dell'RBA di log è un valore di FFFFFFFFFFFFFFFF (se sono in uso RBA a 6 byte) o FFFFFFFFFFFFFFFF (se sono in uso RBA a 8 byte).

Il gestore code emette messaggi [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031De](#) [CSQJ032E](#) per indicare che l'intervallo di log utilizzato è significativo e che è necessario pianificare un'azione per evitare un'interruzione non pianificata.

Il gestore code termina con il codice motivo [00D10257](#) quando il valore RBA raggiunge FFF800000000 (se sono in uso RBA di log a 6 byte) o FFFFFFFC00000000 (se sono in uso RBA di log a 8 byte).

Se gli RBA di log a 6 byte sono in uso, considerare la conversione del gestore code per utilizzare gli RBA di log a 8 byte invece di reimpostare il log del gestore code, seguendo il processo descritto in [“Implementazione dell'indirizzo di byte relativo del log più grande”](#) a pagina 470. La conversione di un gestore code per utilizzare gli RBA di log a 8 byte richiede un'interruzione più breve rispetto alla reimpostazione del log e aumenta il periodo di tempo prima di dover reimpostare il log.

Il messaggio [CSQJ034I](#), emesso durante l'inizializzazione del gestore code, indica la fine dell'intervallo RBA di log per il gestore code come configurato e può essere utilizzato per stabilire se gli RBA di log a 6 o 8 byte sono in uso.

La procedura da seguire per reimpostare il log del gestore code è la seguente:

1. Risolvere eventuali unità di lavoro non risolte. Il numero di unità di lavoro non risolte viene visualizzato all'avvio del gestore code nel messaggio [CSQR005I](#) come conteggio [INDOUBT](#). Ad ogni punto di controllo e all'arresto del gestore code, il gestore code immette automaticamente il comando

**DISPLAY CONN(\*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED)** per fornire informazioni sulle unità di lavoro non risolte.

Consultare [Come vengono risolte le unità di ripristino in dubbio](#) per informazioni sulla risoluzione delle unità di ripristino. L'ultimo ricorso consiste nell'utilizzare il comando MQSC **RESOLVE INDOUBT** per risolvere manualmente le unità di ripristino in dubbio.

2. Chiudere il gestore code in maniera pulita.

È possibile utilizzare **STOP QMGR** o **STOP QMGR MODE(FORCE)** come entrambi questi comandi per eliminare tutte le pagine modificate dai pool di buffer alle serie di pagine.

3. Se un gestore code fa parte di un gruppo di condivisione code, eseguire i backup CFSTRUCT su altri gestori code per tutte le strutture nel gruppo di condivisione code. Ciò garantisce che i backup più recenti non si trovino in questo log del gestore code e che il log di questo gestore code non sia richiesto per il ripristino CFSTRUCT.
4. Definire nuovi log e BSDS utilizzando [CSQJU003](#) (consultare [Programma di utilità di inventario del log delle modifiche](#) per ulteriori informazioni sull'utilizzo del programma di utilità di inventario del log delle modifiche).
5. Eseguire **CSQUTIL RESETPAGE** su tutte le serie di pagine per questo gestore code (per ulteriori informazioni sull'utilizzo di questa funzione, consultare [Copia di una pagina e reimpostazione del log](#)).  
Notare che gli RBA della serie di pagine possono essere reimpostati in modo indipendente, quindi è possibile inoltrare più lavori simultanei (ad esempio, uno per serie di pagine) per ridurre il tempo trascorso per questo passo.
6. Riavviare il gestore code

### Concetti correlati

[“Implementazione dell'indirizzo di byte relativo del log più grande”](#) a pagina 470

Prima di IBM MQ for z/OS 8.0, IBM MQ for z/OS utilizzava un RBA di log di 6 byte per identificare l'ubicazione dei dati all'interno del log. Da IBM MQ for z/OS 8.0, l'RBA del log può essere lungo 8 byte, aumentando il periodo di tempo prima di dover reimpostare il log.

### **z/OS Implementazione dell'indirizzo di byte relativo del log più grande**

Prima di IBM MQ for z/OS 8.0, IBM MQ for z/OS utilizzava un RBA di log di 6 byte per identificare l'ubicazione dei dati all'interno del log. Da IBM MQ for z/OS 8.0, l'RBA del log può essere lungo 8 byte, aumentando il periodo di tempo prima di dover reimpostare il log.

Per i gestori code di IBM MQ 9.2.0 Long Term Support e per i gestori code Continuous Delivery creati prima di IBM MQ 9.2.4, questa funzione deve essere esplicitamente abilitata.

**V 9.2.5** Per i gestori code creati in IBM MQ 9.2.5 e versioni successive, questa funzione è già abilitata.

Consultare [Pianificazione dell'aumento dell'intervallo di log indirizzabile massimo](#) per considerazioni relative all'abilitazione di RBA di log a 8 byte.

**Importante:** **V 9.2.0** Se il gestore code non fa parte di un gruppo di condivisione code e successivamente si esegue la migrazione a IBM MQ for z/OS 9.0.0, avendo abilitato l'RBA di log a 8 byte, assicurarsi di utilizzare **OPMODE=NEWFUNC, 900** in tali release, altrimenti il gestore code non verrà avviato.

Eseguire queste istruzioni, nell'ordine mostrato, per abilitare l'RBA di log di 8 byte su un singolo gestore code IBM MQ for z/OS :

1. **V 9.2.0** Se il gestore code si trova in un gruppo di condivisione code, assicurarsi che tutti i gestori code nel gruppo di condivisione code IBM MQ for z/OS 9.0.0 siano in esecuzione con **OPMODE=(NEWFUNC,900)** prima di passare al passo "2" a pagina 470.

Non è necessario eseguire un'interruzione totale del gruppo di condivisione code per eseguire questa operazione. È possibile arrestare a turno ogni gestore code IBM MQ for z/OS 9.0.0 , modificarlo in **OPMODE=(NEWFUNC,900)** e riavviarlo.

Dopo che tutti i gestori code IBM MQ for z/OS 9.0.0 nel gruppo di condivisione code sono in esecuzione con **OPMODE=(NEWFUNC,900)**, effettuare le seguenti operazioni per ogni gestore code nel gruppo di condivisione code fino a quando tutti i gestori code sono in esecuzione con il nuovo BSDS.

2. Assegnare nuovi dataset BSDS con attributi simili al BSDS corrente. È possibile personalizzare l'esempio CSQ4BSDS ed eliminare qualsiasi istruzione non pertinente oppure è possibile utilizzare il JCL esistente, ma modificare il nome BSDS in un valore simile a ++HLQ++.NEW.BSDS01.

#### **Note:**

- a. Controllare gli attributi del nuovo BSDS prima di inoltrare il lavoro per assegnare il nuovo BSDS. L'unico attributo che potrebbe essere modificato è la dimensione di BSDS.
  - b. Il nuovo BSDS contiene più dati che il BSDS corrente, quindi, è necessario assicurarsi che i nuovi dataset siano allocati con spazio disponibile sufficiente. Il JCL di esempio in thlqual.SCSQPROC(CSQ4BSDS) contiene i valori consigliati quando si definisce un nuovo BSDS.
3. Chiudere il gestore code in maniera pulita.
  4. Eseguire [Programma di utilità di conversione BSDS \(CSQJUCNV\)](#) per convertire il BSDS esistente nei nuovi dataset BSDS. L'esecuzione di questa operazione richiede di solito alcuni secondi.  
Il BSDS esistente non verrà modificato durante questo processo ed è possibile utilizzarlo per l'inizializzazione del gestore code nel caso di una conversione non riuscita.
  5. Ridenominare il BSDS corrente in modo che diventi il vecchio BSDS e il nuovo BSDS in modo che diventi il BSDS corrente, in modo che i nuovi dataset vengano utilizzati al successivo riavvio del gestore code. È possibile utilizzare il comando DFSMS Access Method Services ALTER, ad esempio:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

Accertarsi di immettere anche i comandi per ridenominare sia la parte di dati che quella di indice del cluster VSAM.

6. Riavviare il gestore code. Dovrebbe iniziare nello stesso periodo di tempo che avrebbe fatto quando si utilizza RBA di log a 6 byte.

Se il gestore code non si riavvia correttamente a causa di un errore di accesso al BSDS convertito, tentare di identificare la causa dell'errore, risolvere il problema e ritentare l'operazione. Se necessario, contattare il centro di supporto IBM per assistenza.

Se necessario, la modifica può essere ripristinata a questo punto:

- a. Rinominare il BSDS corrente per diventare il nuovo BSDS.
- b. Ridenominare il vecchio BSDS per diventare il BSDS corrente.
- c. Riavvio del gestore code.

Una volta riavviato correttamente il gestore code con il BSDS convertito, non tentare di avviare il gestore code utilizzando il vecchio BSDS.

7. Il messaggio [CSQJ034I](#) viene emesso durante l'inizializzazione del gestore code per indicare la fine dell'RBA di log per il gestore code come configurato. Confermare che la fine dell'intervallo RBA di log visualizzato sia FFFFFFFFFFFFFFFFFF. Ciò indica che l'RBA di log a 8 byte è in uso.

**Nota:** Per abilitare un RBA di log a 8 byte su un nuovo gestore code, prima che venga avviato, è necessario creare un BSDS di formato versione 1 vuoto e utilizzarlo come input per il programma di utilità di conversione BSDS per produrre un BSDS di formato versione 2. Consultare [Creazione dei dataset di log e di avvio](#) per informazioni su come eseguire questo processo.

### Concetti correlati

[Indirizzo byte relativo log più grande](#)

### Attività correlate

[Pianificazione per aumentare l'intervallo di log indirizzabile massimo](#)

### Riferimenti correlati

[Programma di utilità di conversione BSDS \(CSQJUCNV\)](#)

## **Gestione del dataset di bootstrap (BSDS)**

BSDS (bootstrap data set) viene utilizzato per fare riferimento a dataset di log e record di log. Utilizzare questo argomento per comprendere come esaminare, modificare e ripristinare il BSDS.

Per ulteriori informazioni, consultare [Il dataset bootstrap](#).

Questo argomento descrive le attività coinvolte nella gestione del dataset bootstrap. Contiene le seguenti sezioni:

- [“Scoprire cosa contiene il BSDS” a pagina 471](#)
- [“Modifica di BSDS” a pagina 473](#)
- [“Ripristino di BSDS” a pagina 477](#)

## **Scoprire cosa contiene il BSDS**

È possibile utilizzare il programma di utilità di stampa della mappa del log (CSQJU004) per esaminare i contenuti di BSDS.

Il programma di utilità di stampa della mappa del log (CSQJU004) è un programma di utilità batch che elenca le informazioni memorizzate in BSDS. Per istruzioni sull'esecuzione, consultare [Programma di utilità di stampa della mappa di log](#).

BSDS contiene:

- [Data / ora](#)
- [Stato dataset log attivo](#)

### **Timestamp nel BSDS**

L'output del programma di utilità di stampa della mappa del log mostra la data e l'ora, che vengono utilizzate per registrare la data e l'ora di vari eventi di sistema, memorizzati in BSDS.

Le seguenti date / ore sono incluse nella sezione intestazione del report:

#### **Data/ora del sistema**

Riflette la data e l'ora dell'ultimo aggiornamento del BSDS. La data/ora BSDS può essere aggiornata quando:

- Il gestore code viene avviato.
- La soglia di scrittura viene raggiunta durante le attività di scrittura del log. A seconda del numero di buffer di output specificato e della velocità di attività del sistema, BSDS potrebbe essere aggiornato diverse volte al secondo o potrebbe non essere aggiornato per diversi secondi, minuti o anche ore. Per i dettagli della soglia di scrittura, consultare il parametro WRTHRSR della macro CSQ6LOGP in [Using CSQ6LOGP](#).
- IBM MQ passa in una modalità BSDS singola dalla modalità BSDS doppia normale a causa di un errore. Ciò può verificarsi quando una richiesta di acquisizione, inserimento, riferimento, aggiornamento o eliminazione di un record BSDS ha esito negativo. Quando si verifica questo errore, IBM MQ aggiorna la data / ora nel BSDS rimanente per forzare una mancata corrispondenza della data / ora con il BSDS disabilitato.

#### **DATA/ORA PROGRAMMA DI UTILITÀ**

La data e l'ora in cui il contenuto di BSDS è stato modificato dal programma di utilità di inventario della registrazione modifiche (CSQJU003).

Le seguenti date / ore sono incluse nella parte del report relativa ai dataset di log attivi e di archivio:

#### **Data log attivo**

La data in cui è stata creata la voce del log attivo in BSDS, ovvero quando è stato eseguito il NEWLOG CSQJU003 .

#### **Tempo log attivo**

L'ora in cui è stata creata la voce del log attivo in BSDS, ossia quando è stato eseguito il NEWLOG CSQJU003 .

#### **Data log di archivio**

La data in cui è stata creata la voce del log di archiviazione in BSDS, ovvero quando è stato eseguito il NEWLOG CSQJU003 o l'archivio stesso.

#### **Ora log di archivio**

L'ora in cui è stata creata la voce del log di archiviazione in BSDS, ossia quando è stato eseguito il NEWLOG CSQJU003 o l'archivio stesso.

### **Stato dataset log attivo**

BSDS registra lo stato di un dataset di log attivo come uno dei seguenti:

#### **NUOVA**

Il dataset è stato definito ma non è mai stato utilizzato da IBM MQ oppure il log è stato troncato a un punto prima che il dataset fosse utilizzato per la prima volta. In entrambi i casi, i valori RBA iniziali e finali del dataset vengono reimpostati su zero.

### **Riutilizzabile**

Il dataset è stato definito ma non è mai stato utilizzato da IBM MQ oppure il dataset è stato scaricato. Nell'output della mappa di log di stampa, il valore RBA iniziale per l'ultimo dataset REUSABLE è uguale al valore RBA iniziale dell'ultimo dataset di log di archiviazione.

### **NON RIUTILIZZABILE**

Il dataset contiene record di cui non è stato eseguito l'offload.

### **ARRESTATO**

Il processore offload ha rilevato un errore durante la lettura di un record e non è stato possibile ottenere tale record dall'altra copia del log attivo.

### **Troncato**

Le alternative sono:

- Si è verificato un errore I/O e IBM MQ ha interrotto la scrittura su questo dataset. Il data set del log attivo viene scaricato, iniziando con l'RBA iniziale e continuando fino all'ultimo segmento di record valido nel data set del log attivo troncato. L'RBA dell'ultimo segmento di record valido è inferiore all'RBA finale del dataset di log attivo. La registrazione viene passata al successivo dataset di log attivo disponibile e continua senza interruzioni.

oppure

- È stata richiamata una funzione ARCHIVE LOG, che ha troncato il log attivo.

Lo stato viene visualizzato nell'emissione del programma di utilità di stampa della mappa del log.

## **Modifica di BSDS**

Non è necessario eseguire operazioni speciali per mantenere aggiornato il BSDS con i record degli eventi di registrazione perché IBM MQ lo fa automaticamente.

Tuttavia, è possibile modificare il BSDS se si esegue una delle seguenti operazioni:

- Aggiungere altri dataset di log attivi.
- Copiare i dataset di log attivi in dataset appena assegnati, ad esempio, quando si forniscono allocazioni di log attivi più grandi.
- Spostare i dataset di log su altri dispositivi.
- Recuperare un BSDS danneggiato.
- Eliminare i dataset di log di archivio obsoleti.

È possibile modificare BSDS eseguendo il programma di utilità di inventario del log delle modifiche (CSQJU003). Eseguire questo programma di utilità solo quando il gestore code è inattivo o si potrebbero ottenere risultati incongruenti. L'azione del programma di utilità è controllata da istruzioni nel dataset SYSIN. Questa sezione mostra diversi esempi. Per istruzioni complete, vedere [Programma di utilità di inventario del log delle modifiche](#).

È possibile copiare un dataset del log attivo solo quando il gestore code è inattivo poiché IBM MQ assegna i dataset del log attivo come esclusivi (DISP = OLD) all'avvio del gestore code.

## **Modifiche per i log attivi**

Utilizzare questo argomento per comprendere come è possibile modificare i log attivi utilizzando BSDS.

È possibile aggiungere, eliminare e registrare voci nel BSDS per i log attivi utilizzando il programma di utilità di registrazione delle modifiche. Qui vengono mostrati solo degli esempi; sostituire i nomi dei dataset mostrati con quelli che si desidera utilizzare. Per ulteriori dettagli sul programma di utilità, consultare [Programma di utilità di inventario del log delle modifiche](#).

Consultare queste sezioni per ulteriori informazioni:

- [Aggiunta di voci record a BSDS](#)
- [Eliminazione delle informazioni sul dataset di log attivo dal BSDS](#)
- [Registrazione delle informazioni sul dataset di log in BSDS](#)

- Aumento della dimensione del log attivo
- L'utilizzo di CSQJUFMT

### **Aggiunta di voci record a BSDS**

Se un log attivo è stato contrassegnato come "arrestato", non viene riutilizzato per la registrazione; tuttavia, continua ad essere utilizzato per la lettura. Utilizzare i servizi del metodo di accesso per definire nuovi dataset di log attivi, quindi utilizzare il programma di utilità di inventario del log di modifiche per registrare i nuovi dataset in BSDS. Ad esempio, utilizzare:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSNAMES=MQM111.LOGCOPY2.DS10,COPY2
```

Se si sta copiando il contenuto di un vecchio dataset di log attivo nel nuovo, è anche possibile fornire l'intervallo RBA e le date / ore di inizio e fine sulla funzione NEWLOG.

### **Eliminazione delle informazioni sul dataset del log attivo da BSDS**

Per eliminare informazioni su un dataset di log attivo da BSDS, è possibile utilizzare:

```
DELETE DSNAMES=MQM111.LOGCOPY1.DS99
DELETE DSNAMES=MQM111.LOGCOPY2.DS99
```

### **Registrazione delle informazioni sul dataset di log in BSDS**

Per registrare le informazioni su un dataset di log attivo esistente in BSDS, utilizzare:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

Potrebbe essere necessario inserire un record contenente questo tipo di informazioni in BSDS perché:

- La voce per il data set è stata eliminata, ma è di nuovo necessaria.
- Si sta copiando il contenuto di un dataset di log attivo in un altro dataset.
- Si sta ripristinando BSDS da una copia di backup.

### **Aumento della dimensione del log attivo**

Esistono due metodi per realizzare questo processo.

1. Quando il gestore code è attivo:
  - a. Definire nuovi dataset di log più grandi utilizzando JCL.
  - b. Aggiungere i nuovi dataset di log al gestore code attivo utilizzando il comando MQSC DEFINE LOG.
  - c. Utilizzare il comando MQSC ARCHIVE LOG per spostare il log attivo corrente, in modo che sia un nuovo log più grande.
  - d. Attendere il completamento dell'archivio del dataset del log attivo più piccolo.
  - e. Chiudere il gestore code, utilizzando il programma di utilità CSQJU003 per rimuovere i vecchi piccoli log attivi.
  - f. Riavviare il gestore code.
2. Quando il gestore code non è attivo:

- a. Chiudere il gestore code. Questo passo è richiesto perché IBM MQ assegna tutti i dataset di log attivi per il suo utilizzo esclusivo quando è attivo.
- b. Utilizzare Access Method Services ALTER con l'opzione NEWNAME per ridenominare i dataset di log attivi.
- c. Utilizzare Access Method Services DEFINE per definire dataset di log attivi più grandi.  
 Riutilizzando i vecchi nomi di dataset, non è necessario eseguire il programma di utilità di inventario del log di modifica per stabilire nuovi nomi nei BSDS. I vecchi nomi dataset e gli intervalli RBA corretti sono già presenti nei BSDS.
- d. Utilizzare Access Method Services REPRO per copiare i vecchi dataset (ridenominati) nei nuovi dataset appropriati.  
**Nota:** Questo passo può richiedere molto tempo, quindi l'azienda potrebbe non essere in azione per questo periodo.
- e. Avviare il gestore code.

Se tutti i dataset di log hanno la stessa dimensione, il sistema sarà operativo più coerente ed efficiente. Se i dataset di log non hanno la stessa dimensione, è più difficile tenere traccia dei log del sistema e quindi lo spazio può essere sprecato.

### L'uso di CSQJUFMT

Non eseguire un formato CSQJUFMT quando si aumenta la dimensione di un log attivo.

Se si esegue CSQJUFMT (per migliorare le prestazioni la prima volta che il gestore code scrive nel nuovo log attivo) si ricevono i seguenti messaggi:

```
IEC070I 203-204,XS95GTLX,REPRO02,OUTPUT,B857,SPMG02, 358
IEC070I MG.W.MG4E.LOGCOPY1.DS02,MG.W.MG4E.LOGCOPY1.DS02.DATA,
IDC3302I ACTION ERROR ON MG.W.MG4E.LOGCOPY1.DS02
IDC3351I ** VSAM I/O RETURN CODE IS 28 - RPLFDBWD = X'2908001C'
IDC31467I MAXIMUM ERROR LIMIT REACHED.

IDC0005I NUMBER OF RECORDS PROCESSED WAS 0
```

Inoltre, se si utilizza Access Method Services REPRO, assicurarsi di definire un nuovo log vuoto.

Se si utilizza REPRO per copiare il vecchio dataset (ridenominato) nel rispettivo nuovo dataset, il valore predefinito è NOREPLACE.

Ciò significa che REPRO non sostituisce un record che si trova già nel dataset designato. Quando la formattazione viene eseguita sul dataset, il valore RBA viene reimpostato. Il risultato della rete è un dataset che non è vuoto dopo la formattazione.

### Modifiche per i log di archiviazione

Utilizzare questo argomento per comprendere come modificare i log di archivio.

È possibile aggiungere, eliminare e modificare le password delle voci in BSDS per i log di archivio. Qui vengono mostrati solo esempi; sostituire i nomi dei dataset mostrati con quelli che si desidera utilizzare. Per ulteriori dettagli sul programma di utilità, consultare [Il programma di utilità di inventario del log delle modifiche](#).

- [Aggiunta di un log di archivio](#)
- [Eliminazione di un log di archivio](#)
- [Modifica della password di un log di archivio](#)

### Aggiunta di un log di archivio

Quando il ripristino di un oggetto dipende dalla lettura di un dataset di log di archiviazione esistente, BSDS deve contenere informazioni su tale dataset in modo che IBM MQ possa trovarlo. Per registrare informazioni su un dataset di log di archiviazione esistente in BSDS, utilizzare:

```
NEWLOG DSN=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

### Eliminazione di un log di archivio

Per eliminare un intero dataset di log di archivio su uno o più volumi, utilizzare:

```
DELETE DSN=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

### Modifica della parola d'ordine di un log di archivio

Se si modifica la password di un dataset di log di archivio esistente, è necessario modificare anche le informazioni in BSDS.

1. Elencare il BSDS, utilizzando il programma di utilità di stampa della mappa del log.
2. Eliminare la voce per il dataset del log di archiviazione con la password modificata, utilizzando la funzione DELETE del programma di utilità CSQJU003 (consultare l'argomento [Programma di utilità di inventario del log di modifica](#)).
3. Denominare il dataset come per un nuovo dataset del log di archiviazione. Utilizzare la funzione NEWLOG del programma di utilità CSQJU003 (consultare l'argomento [Programma di utilità di inventario del log delle modifiche](#)) e fornire la nuova password, gli RBA iniziali e finali e i numeri di serie del volume (disponibili nell'output del programma di utilità di associazione del log di stampa, consultare [Programma di utilità di associazione del log di stampa](#)).

Per modificare la password per i nuovi dataset di log di archiviazione, utilizzare:

```
ARCHIVE PASSWORD= password
```

Per interrompere l'inserimento delle password nei nuovi dataset di log di archiviazione, utilizzare:

```
ARCHIVE NOPASSWD
```

**Nota:** Utilizzare la funzione di utilità ARCHIVE solo se non si dispone di un gestore della sicurezza esterno.

### Modifica del qualificatore di alto livello (HLQ) per i log e BSDS

Utilizzare questo argomento per comprendere la procedura richiesta per modificare il qualificatore di alto livello (HLQ).

### Prima di iniziare

È necessario terminare il gestore code normalmente prima di copiare i log o i dataset nei nuovi dataset. Questo per garantire che i dati siano congruenti e che non sia necessario alcun ripristino durante il riavvio.

### Informazioni su questa attività

Questa attività fornisce informazioni su come modificare l'HLQ per i log e BSDS. A tale scopo, attenersi alla seguente procedura:

## Procedura

1. Eseguire il programma di utilità di stampa del log CSQJU004 per registrare le informazioni sul dataset del log. Queste informazioni sono necessarie in seguito.
2. È possibile:
  - a) eseguire il backup e il ripristino DSS con ridenominazione sul log e sui dataset BSDS da ridenominare oppure
  - b) utilizzare AMS DEFINE e REPRO per creare i dataset HLQ e copiare i dati dai vecchi dataset.
3. Modificare le procedure MSTR e CHIN per puntare ai nuovi dataset.
4. Eliminare le informazioni di log precedenti nella nuova copia di BSDS utilizzando CSQJU003.
5. Definire i nuovi dataset di log per il nuovo BSDS utilizzando la funzione NEWLOG di CSQJU003. Conservare tutte le informazioni su ciascun log, ad eccezione dell'HLQ.
6. Il nuovo BSDS deve riflettere le stesse informazioni registrate per i vecchi log nel vecchio BSDS. L'HLQ dovrebbe essere l'unica cosa che è cambiata.

## Operazioni successive

Confrontare l'output CSQJU004 per il vecchio e il nuovo BSDS per assicurarsi che siano ESATTAMENTE uguali (ad eccezione degli HLQ) prima di avviare il gestore code.

**Nota:** È necessario prestare attenzione quando si eseguono queste operazioni. Azioni non corrette potrebbero portare a situazioni irrecuperabili. Controllare l'output di PRINT LOG MAP UTILITY e verificare che tutte le informazioni necessarie per il recupero o il riavvio siano state incluse.

### Ripristino di BSDS

Se IBM MQ sta funzionando in modalità BSDS doppia e un BSDS si danneggia, forzando IBM MQ in modalità BSDS singola, IBM MQ continua a funzionare senza problemi (fino al successivo riavvio).

Per riportare l'ambiente alla modalità BSDS doppia:

1. Utilizzare Access Method Services per ridenominare o eliminare il BSDS danneggiato e per definire un nuovo BSDS con lo stesso nome del BSDS danneggiato. È possibile trovare istruzioni di controllo di esempio nel lavoro CSQ4BREC in thlqual.SCSQPROC.
2. Immettere il comando IBM MQ RECOVER BSDS per creare una copia del BSDS valido nel dataset appena assegnato e per ripristinare la modalità BSDS doppia.

Se IBM MQ funziona in modalità BSDS singola e BSDS è danneggiato o se IBM MQ funziona in modalità BSDS doppia ed entrambi i BSDS sono danneggiati, il gestore code si arresta e non si riavvia fino a quando non vengono riparati i dataset BSDS. In questo caso:

1. Individuare il BSDS associato al dataset di log di archiviazione più recente. Il nome del dataset del log di archiviazione più recente viene visualizzato nel log del lavoro nell'ultima ricorrenza del messaggio CSQJ003I, che indica che l'elaborazione dell'offload è stata completata correttamente. In preparazione per il resto di questa procedura, è buona norma conservare un registro di tutti gli archivi di successo annotati da quel messaggio:
  - Se i log di archivio sono su DASD, il BSDS viene assegnato su qualsiasi DASD disponibile. Il nome BSDS è simile al nome del dataset del log di archivio corrispondente; modificare solo la prima lettera dell'ultimo qualificatore, da A a B, come in questo esempio:

#### Nome log di archivio

CSQ.ARCHLOG1. **A** 0000001

#### Nome copia BSDS

CSQ.ARCHLOG1. **B** 0000001

- Se i log di archivio sono su nastro, BSDS è il primo dataset del primo volume di log di archiviazione. BSDS non viene ripetuto su volumi successivi.

2. Se il dataset di log di archivio più recente non ha alcuna copia di BSDS (ad esempio, perché si è verificato un errore durante l'offload), individuare una copia precedente di BSDS da un'elaborazione di offload precedente.
3. Ridenominare i BSDS *danneggiati* utilizzando il comando ALTER di Access Method Services con l'opzione NEWNAME. Se si desidera eliminare un BSDS danneggiato, utilizzare il comando DELETE di Access Method Services. Per ogni BSDS danneggiato, utilizzare Access Method Services per definire un nuovo BSDS come dataset di sostituzione. Il lavoro CSQ4BREC in thlqual.SCSQPROC contiene le istruzioni di controllo di Access Method Services per definire un nuovo BSDS.
4. Utilizzare il comando REPRO di Access Method Services per copiare il BSDS dal log di archiviazione in uno dei BSDS di sostituzione definiti al passo "3" a [pagina 478](#). Non copiare alcun dato nel secondo BSDS di sostituzione, come nel passo "5" a [pagina 479](#).

a. Stampare il contenuto del BSDS di sostituzione.

Utilizzare il programma di utilità di stampa della mappa del log (CSQJU004) per stampare il contenuto del BSDS di sostituzione. Ciò consente di esaminare il contenuto del BSDS di sostituzione prima di continuare il lavoro di recupero.

b. Aggiornare l'inventario del dataset del log di archiviazione nel BSDS di sostituzione.

Esaminare l'output del programma di utilità di stampa della mappa del log e controllare che il BSDS di sostituzione non contenga un record del log di archivio da cui è stato copiato il BSDS. Se il BSDS di sostituzione è una copia obsoleta, il relativo inventario potrebbe non contenere tutti i dataset di log di archivio creati più di recente. L'inventario BSDS dei dataset del log di archivio deve essere aggiornato per riflettere l'inventario del sottosistema corrente.

Utilizzare l'istruzione NEWLOG del programma di utilità di modifica inventario log (CSQJU003) per aggiornare il BSDS di sostituzione, aggiungendo un record del log di archiviazione da cui è stato copiato il BSDS. Se il dataset del log di archivio è protetto da password, utilizzare l'opzione PASSWORD della funzione NEWLOG. Inoltre, se il dataset del log di archivio è catalogato, assicurarsi che l'opzione CATALOG della funzione NEWLOG sia impostata correttamente su CATALOG=YES. Utilizzare l'istruzione NEWLOG per aggiungere ulteriori dataset di log di archiviazione creati successivamente alla copia BSDS.

c. Aggiornare le password nel BSDS di sostituzione.

BSDS contiene le password per i dataset di log di archiviazione e per i dataset di log attivi. Per assicurarsi che le password nel BSDS di sostituzione riflettano le password correnti utilizzate dalla propria installazione, utilizzare la funzione di utilità ARCHIVE del log di modifica con l'opzione PASSWORD.

d. Aggiornare l'inventario del dataset di log attivo nel BSDS di sostituzione.

In circostanze insolite, l'installazione potrebbe aver aggiunto, eliminato o rinominato i dataset di log attivi da quando è stato copiato BSDS. In questo caso, il BSDS di sostituzione non riflette il numero o i nomi effettivi dei dataset di log attivi attualmente in uso nell'installazione.

Se è necessario eliminare un dataset del log attivo dall'inventario del log BSDS di sostituzione, utilizzare la funzione DELETE del programma di utilità di inventario del log di modifica.

Se è necessario aggiungere un dataset di log attivo all'inventario di log BSDS di sostituzione, utilizzare la funzione NEWLOG del programma di utilità di inventario del log di modifica. Assicurarsi che l'intervallo RBA sia specificato correttamente nella funzione NEWLOG. Se il dataset del log attivo è protetto da password, utilizzare l'opzione PASSWORD.

Se è necessario ridenominare un dataset di log attivo nell'inventario di log BSDS di sostituzione, utilizzare la funzione DELETE del programma di utilità di inventario del log di modifica, seguita dalla funzione NEWLOG. Assicurarsi che l'intervallo RBA sia specificato correttamente nella funzione NEWLOG. Se il dataset del log attivo è protetto da password, utilizzare l'opzione PASSWORD.

e. Aggiornare gli intervalli di RBA del log attivo nel BSDS di sostituzione.

Successivamente, quando il gestore code viene riavviato, confronta gli RBA dei dataset di log attivi elencati in BSDS con gli RBA trovati nei dataset di log attivi effettivi. Se gli RBA non concordano, il gestore code non viene riavviato. Il problema viene ingrandito quando si utilizza una vecchia copia

di BSDS. Per risolvere questo problema, utilizzare il programma di utilità di inventario del log di modifica (CSQJU003) per regolare gli RBA trovati in BSDS utilizzando gli RBA nei dataset di log attivi effettivi. Eseguire questa operazione per:

- Utilizzo del programma di utilità Stampa record di log (CSQ1LOGP) per stampare un report di riepilogo del dataset di log attivo. Mostra gli RBA iniziali e finali.
- Confronto degli intervalli RBA effettivi con gli intervalli RBA appena stampati, quando sono noti gli RBA di tutti i dataset di log attivi.

Se gli intervalli RBA sono uguali per tutti i dataset di log attivi, è possibile procedere al passo di ripristino successivo senza ulteriore lavoro.

Se gli intervalli RBA non sono uguali, regolare i valori in BSDS per riflettere i valori effettivi. Per ogni dataset di log attivo che deve avere l'intervallo RBA regolato, utilizzare la funzione DELETE del programma di utilità di inventario del log di modifiche per cancellare il dataset di log attivo dall'inventario nel BSDS di sostituzione. Utilizzare quindi la funzione NEWLOG per ridefinire il dataset di log attivo in BSDS. Se i dataset di log attivi sono protetti da password, utilizzare l'opzione PASSWORD della funzione NEWLOG.

- f. Se vengono specificati solo due dataset di log attivi per ogni copia del log attivo, IBM MQ può avere difficoltà durante il riavvio del gestore code. Il problema può verificarsi quando uno dei dataset di log attivi è pieno e non è stato scaricato, mentre il secondo dataset di log attivo è vicino al riempimento. In questo caso, aggiungere un nuovo dataset del log attivo per ciascuna copia del log attivo e definire ogni nuovo dataset del log attivo nell'inventario del log BSDS di sostituzione.

Utilizzare il comando DEFINE di Access Method Services per definire un nuovo dataset di log attivo per ogni copia del log attivo e utilizzare la funzione NEWLOG del programma di utilità per l'inventario del log di modifiche per definire i nuovi dataset di log attivo nel BSDS di sostituzione. Non è necessario specificare gli intervalli RBA nell'istruzione NEWLOG. Tuttavia, se i dataset di log attivi sono protetti da password, utilizzare l'opzione PASSWORD della funzione NEWLOG. Le istruzioni di controllo di esempio per eseguire questa attività sono disponibili nel lavoro CSQ4LREC in thlqual.SCSQPROC.

5. Copiare il BSDS aggiornato nel secondo nuovo dataset BSDS. I BSDS sono ora identici.

Utilizzare il programma di utilità di stampa della mappa del log (CSQJU004) per stampare il contenuto del secondo BSDS di sostituzione a questo punto.

6. Consultare [Problemi del log attivo](#) per informazioni sulle operazioni da eseguire se si è perso il dataset del log attivo corrente.
7. Riavviare il gestore code utilizzando il BSDS appena creato. IBM MQ determina l'RBA corrente e quali log attivi devono essere archiviati.

## **Gestione delle serie di pagine**

Utilizzare questo argomento per comprendere come gestire le serie di pagine associate a un gestore code.

Questo argomento descrive come aggiungere, copiare e generalmente gestire le serie di pagine associate a un gestore code. Contiene le seguenti sezioni:

- [“Come modificare il qualificatore di alto livello \(HLQ\) per le serie di pagine” a pagina 480](#)
- [“Come aggiungere una serie di pagine a un gestore code” a pagina 480](#)
- [“Cosa fare quando una delle serie di pagine diventa piena” a pagina 480](#)
- [“Come bilanciare i carichi sui set di pagine” a pagina 481](#)
- [Come aumentare la dimensione di una serie di pagine](#)
- [“Come ridurre un set di pagine” a pagina 485](#)
- [“Come reintrodurre una serie di pagine” a pagina 485](#)

- [“Come eseguire il backup e il ripristino delle serie di pagine” a pagina 486](#)
- [“Come eliminare le serie di pagine” a pagina 490](#)
- [“Come eseguire il backup e ripristinare le code utilizzando CSQUTIL” a pagina 490](#)

Consultare [Serie di pagine](#) per una descrizione delle serie di pagine, delle classi di memoria, dei buffer e dei pool di buffer e per alcune delle considerazioni sulle prestazioni applicabili.

## Come modificare il qualificatore di alto livello (HLQ) per le serie di pagine

Questa attività fornisce informazioni su come modificare l'HLQ per le serie di pagine. Per eseguire questa attività, effettuare le seguenti operazioni:

1. Definire le nuove serie di pagine HLQ.
2. Se l'assegnazione della dimensione è la stessa delle vecchie serie di pagine, copiare la serie di pagine esistente utilizzando REPRO nelle nuove serie di pagine HLQ vuote.
3. Se si sta aumentando la dimensione delle serie di pagine, utilizzare la funzione FORMAT di CSQUTIL per formattare le pagine di destinazione e quindi la funzione COPYPAGE di CSQUTIL per copiare tutti i messaggi dalla serie di pagine di origine alla serie di pagine di destinazione.

Per ulteriori informazioni, consultare [Formattazione delle serie di pagine \(FORMAT\)](#) e [Espansione di una serie di pagine \(COPYPAGE\)](#).

4. Modificare l'istruzione DD CSQP00xx nella procedura del gestore code per puntare alle nuove serie di pagine HLQ.

Riavviare il gestore code e verificare le modifiche alle serie di pagine.

## Come aggiungere una serie di pagine a un gestore code

Questa descrizione presuppone che si disponga di un gestore code già in esecuzione. Potrebbe essere necessario aggiungere una serie di pagine se, ad esempio, il gestore code deve gestire nuove applicazioni utilizzando nuove code.

Per aggiungere una nuova serie di pagine, utilizzare la seguente procedura:

1. Definire e formattare la nuova serie di pagine. È possibile utilizzare il JCL di esempio in `thlqual.SCSQPROC(CSQ4PAGE)` come base. Per ulteriori informazioni, consultare [Formattazione delle serie di pagine \(FORMAT\)](#).

Fare attenzione a non formattare le serie di pagine in uso, a meno che questo non sia ciò che si intende. In tal caso, utilizzare l'opzione FORCE della funzione di utilità FORMAT.

2. Utilizzare il comando DEFINE PSID con l'opzione DSN per associare la serie di pagine a un pool di buffer.
3. Aggiungere le definizioni di classe di memoria appropriate per la serie di pagine immettendo comandi DEFINE STGCLASS.
4. Facoltativamente, per documentare come è configurato il gestore code:
  - a. Aggiungere la nuova serie di pagine alla procedura dell'attività avviata per il gestore code.
  - b. Aggiungere una definizione per la nuova serie di pagine al dataset di inizializzazione CSQINP1 .
  - c. Aggiungere una definizione per la nuova classe di memoria al membro del dataset di inizializzazione CSQ4INJR .

Per dettagli sui comandi DEFINE PSID e DEFINE STGCLASS, consultare [DEFINE PSID](#) e [DEFINE STGCLASS](#).

## Cosa fare quando una delle serie di pagine diventa piena

È possibile ottenere informazioni sull'utilizzo delle serie di pagine utilizzando il comando IBM MQ DISPLAY USAGE. Ad esempio, il comando:

```
DISPLAY USAGE PSID(03)
```

visualizza lo stato corrente della serie di pagine 03. Questo indica quante pagine libere ha questa serie di pagine.

Se sono state definite estensioni secondarie per le serie di pagine, queste vengono espanso dinamicamente ogni volta che si riempiono. Infine, vengono utilizzate tutte le estensioni secondarie oppure non è disponibile ulteriore spazio su disco. Se ciò si verifica, un'applicazione riceve il codice di ritorno MQRC\_STORAGE\_MEDIUM\_FULL.

Se un'applicazione riceve un codice di ritorno di MQRC\_STORAGE\_MEDIUM\_FULL da una chiamata MQI, ciò indica chiaramente che non è rimasto spazio sufficiente nella serie di pagine. Se il problema persiste o è probabile che si ripeta, è necessario fare qualcosa per risolverlo.

È possibile risolvere questo problema in diversi modi:

- Bilanciare il carico tra le serie di pagine spostando le code da una serie di pagine all'altra.
- Espandere la serie di pagine. Per istruzioni, consultare [“Come aumentare la dimensione di un set di pagine”](#) a pagina 483.
- Ridefinire la serie di pagine in modo che possa espandersi oltre i 4 GB ad una dimensione massima di 64 GB. Consultare [Definizione di una serie di pagine con dimensioni superiori a 4 GB](#) per istruzioni.

## Come bilanciare i carichi sui set di pagine

Il bilanciamento del carico sulle serie di pagine significa spostare i messaggi associati a una o più code da una serie di pagine a un'altra, meno utilizzata, serie di pagine. Utilizzare questa tecnica se non è pratico espandere la serie di pagine.

Per identificare quali code utilizzano una serie di pagine, utilizzare i comandi IBM MQ appropriati. Ad esempio, per individuare quali code sono associate alla serie di pagine 02, per prima cosa, individuare le classi di memoria associate alla serie di pagine 02, utilizzando il comando:

```
DISPLAY STGCLASS(*) PSID(02)
```

Quindi, utilizzare il comando riportato di seguito per individuare le code che utilizzano la classe di memoria:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

## Spostamento di una coda non condivisa

Per spostare le code e i relativi messaggi da una serie di pagine all'altra, utilizzare il comando MQSC MOVE QLOCAL (descritto in [MOVE QLOCAL](#)). Una volta identificate la coda o le code che si desidera spostare in una nuova serie di pagine, attenersi alla seguente procedura per ciascuna di queste code:

1. Assicurarsi che la coda che si desidera spostare non sia utilizzata da alcuna applicazione (ovvero, i valori IPPROCS e OPPROCS del comando DISPLAY QSTATUS sono zero) e che non abbia messaggi senza commit (il valore UNCOM del comando DISPLAY QSTATUS è NO).

**Nota:** L'unico modo per garantire che questo stato continui è modificare temporaneamente l'autorizzazione di sicurezza della coda. Per ulteriori informazioni, consultare [Profili per la sicurezza della coda](#).

Se non è possibile eseguire questa operazione, le fasi successive di questa procedura potrebbero avere esito negativo se le applicazioni iniziano ad utilizzare la coda nonostante i passi precauzionali come l'impostazione di PUT (DISABLED). Tuttavia, i messaggi non possono mai essere persi da questa procedura.

2. Impedire alle applicazioni di inserire i messaggi nella coda spostati modificando la definizione della coda per disabilitare MQPUT . Modificare la definizione della coda in PUT (DISABLED).
3. Definire una coda temporanea con gli stessi attributi della coda che si sta spostando, utilizzando il comando:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

**Nota:** Se questa coda temporanea esiste già da una precedente esecuzione, eliminarla prima di eseguire la definizione.

4. Spostare i messaggi nella coda temporanea utilizzando il seguente comando:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Eliminare la coda che si sta spostando, utilizzando il comando:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Definire una nuova classe di memoria che si associ alla serie di pagine richiesta, ad esempio:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Aggiungere la nuova definizione di classe di memoria ai dataset CSQINP2 pronti per il successivo riavvio del gestore code.

7. Ridefinire la coda che si sta spostando, modificando l'attributo della classe di memoria:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
```

Quando la coda viene ridefinita, si basa sulla coda temporanea creata nella fase “3” a pagina 482.

8. Spostare nuovamente i messaggi nella nuova coda, utilizzando il comando:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. La coda creata nel passo “3” a pagina 482 non è più richiesta. Utilizzare il seguente comando per eliminarla:

```
DELETE QL(TEMP_QUEUE)
```

10. Se la coda che viene spostata è stata definita nei dataset CSQINP2 , modificare l'attributo STGCLASS del comando DEFINE QLOCAL appropriato nei dataset CSQINP2 . Aggiungere la parola chiave REPLACE in modo che la definizione della coda esistente venga sostituita.

Figura 37 a pagina 483 mostra un'estrazione da un job di bilanciamento del carico.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
//      DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE STGCLASS(NEW) PSID(2)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) STGCLASS(NEW)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figura 37. Estrai da un job di bilanciamento del carico per una serie di pagine

## Come aumentare la dimensione di un set di pagine

È possibile assegnare inizialmente una serie di pagine superiore a 4 GB, consultare [Definizione di una serie di pagine superiore a 4 GB](#)

Una serie di pagine può essere definita per essere espansa automaticamente quando diventa piena specificando EXPAND (SYSTEM) o EXPAND (USER). Se la serie di pagine è stata definita con EXPAND (NONE), è possibile espanderla in due modi:

- Modificare la sua definizione per consentire l'espansione automatica. Consultare [Modifica di una serie di pagine per consentire l'espansione automatica](#)
- Creare una nuova serie di pagine più grande e copiare i messaggi dalla vecchia serie di pagine alla nuova. Consultare [Spostamento dei messaggi in una nuova serie di pagine più grande](#)

### Definizione di una serie di pagine superiore a 4 GB

IBM MQ può utilizzare una pagina di dimensione fino a 64 GB, purché il dataset sia definito con 'indirizzabilità estesa' a VSAM. L'indirizzabilità estesa è un attributo conferito da una classe di dati SMS.

**Nota:** Le serie di pagine e i dataset di log attivi sono idonei a risiedere nello spazio di indirizzamento esteso (EAS) di un EAV (extended address volume) e, da z/OS V1.12, un dataset di log di archiviazione può risiedere anche in EAS.

Nell'esempio mostrato nel seguente JCL di esempio, la classe di gestione 'EXTENDED' è definita per SMS con 'Extended addressability'. Se la serie di pagine esistente non è attualmente definita come avente indirizzabilità estesa, utilizzare il seguente metodo per migrare a una serie di dati in formato di indirizzabilità estesa.

1. Chiudere il gestore code.
2. Utilizzare Access Method Services per ridenominare la serie di pagine esistente.
3. Definire una serie di pagine di destinazione, della stessa dimensione della serie di pagine esistente, ma con DATACLAS (EXTENDED).

**Nota:** I dataset in formato esteso devono essere gestiti tramite SMS. Questi sono i meccanismi per richiedere il formato esteso per i dataset VSAM:

- Utilizzando una classe di dati che ha un valore DSNTYPE di EXT e il sottoparametro R o P per indicare richiesto o preferito.
- Codifica di DSNTYPE=EXTREQ (è richiesto il formato esteso) o DSNTYPE=EXTPREF (è preferito il formato esteso) sull'istruzione DD.
- Codifica del parametro LIKE= sull'istruzione DD per fare riferimento a un dataset di formato esteso esistente.

Per ulteriori informazioni, consultare [Limitazioni sulla definizione di dataset in formato esteso](#).

4. Utilizzare la funzione COPYPAGE di CSQUTIL per copiare tutti i messaggi dalla serie di pagine di origine alla serie di pagine di destinazione. Per ulteriori dettagli, consultare [Espansione di una serie di pagine \(COPYPAGE\)](#).
5. Riavviare il gestore code.
6. Modificare la serie di pagine per utilizzare l'espansione del sistema, per consentirgli di continuare a crescere oltre la sua allocazione corrente.

Il seguente JCL mostra i comandi Access Method Services di esempio:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER 'VICY.CSQ1.PAGE01' -
NEWNAME('VICY.CSQ1.PAGE01.OLD')
ALTER 'VICY.CSQ1.PAGE01.DATA' -
NEWNAME('VICY.CSQ1.PAGE01.DATA.OLD')
DEFINE CLUSTER (NAME('VICY.CSQ1.PAGE01') -
MODEL('VICY.CSQ1.PAGE01.OLD') -
DATACLAS(EXTENDED))
/*
```

### Modifica di una serie di pagine per consentire l'espansione automatica

Utilizzare il comando ALTER PSID con le opzioni EXPAND (USER) o EXPAND (SYSTEM). Consultare [ALTER PSID](#) e [Espansione di una serie di pagine \(COPYPAGE\)](#) per informazioni generali sull'espansione delle serie di pagine.

### Spostamento dei messaggi in una nuova serie di pagine più grande

Questa tecnica implica l'arresto e il riavvio del gestore code. In questo modo vengono eliminati tutti i messaggi non persistenti che non si trovano nelle code condivise al momento del riavvio. Se si dispone di messaggi non persistenti che non si desidera eliminare, utilizzare invece il bilanciamento del carico. Per ulteriori dettagli, consulta [“Come bilanciare i carichi sui set di pagine”](#) a pagina 481. In questa descrizione, la serie di pagine che si desidera espandere viene indicata come serie di pagine *origine*; la nuova serie di pagine più grande viene indicata come serie di pagine *destinazione*.

Eeguire queste operazioni:

1. Chiudere il gestore code.
2. Definire la serie di pagine di destinazione, verificando che sia più grande della serie di pagine di origine, con un valore di estensione secondaria maggiore.
3. Utilizzare la funzione FORMAT di CSQUTIL per formattare la serie di pagina di destinazione. Per ulteriori dettagli, vedere [Formattazione delle serie di pagine \(FORMAT\)](#).
4. Utilizzare la funzione COPYPAGE di CSQUTIL per copiare tutti i messaggi dalla serie di pagine di origine alla serie di pagine di destinazione. Per ulteriori dettagli, consultare [Espansione di una serie di pagine \(COPYPAGE\)](#).
5. Riavviare il gestore code utilizzando la pagina di destinazione impostata effettuando una delle seguenti operazioni:
  - Modificare la procedura dell'attività avviata del gestore code per fare riferimento alla serie di pagine di destinazione.

- Utilizzare Access Method Services per eliminare la serie di pagine di origine e ridenominare la serie di pagine di destinazione, assegnandole lo stesso nome della serie di pagine di origine.

**Attenzione:**

Prima di eliminare qualsiasi serie di pagine IBM MQ , assicurarsi di aver eseguito le copie di backup richieste.

## Come ridurre un set di pagine

Impedire a tutti gli utenti, diversi dall'amministratore IBM MQ , di utilizzare il gestore code. Ad esempio, modificando le impostazioni di sicurezza di accesso.

Se si dispone di una serie di pagine di grandi dimensioni che è per lo più vuota (come mostrato dal comando DISPLAY USAGE), è possibile ridurre la dimensione. La procedura per eseguire questa operazione implica l'uso delle funzioni COPY, FORMAT e LOAD di CSQUTIL (consultare il programma di utilità [IBM MQ](#) ). Questa procedura non funziona per la serie di pagine zero (0), poiché non è pratico ridurre la dimensione di questa serie di pagine; l'unico modo per farlo è reinizializzare il gestore code (consultare [“Reinizializzazione di un gestore code”](#) a pagina 509 ). Il prerequisito di questa procedura consiste nel tentare di rimuovere tutti gli utenti dal sistema in modo che tutte le UOW siano complete e le serie di pagine siano congruenti.

1. Utilizzare il comando STOP QMGR con l'attributo QUIESCE o FORCE per arrestare il gestore code.
2. Eseguire la funzione SCOPY di CSQUTIL con l'opzione PSID per copiare tutti i dati del messaggio dalla serie di pagine di grandi dimensioni e salvarli in una serie di dati sequenziali.
3. Definire una nuova serie di pagine più piccola per sostituire la serie di pagine di grandi dimensioni.
4. Eseguire la funzione FORMAT TYPE (NEW) di CSQUTIL rispetto alla serie di pagine creata nel passo [“3”](#) a pagina 485.
5. Riavviare il gestore code utilizzando la serie di pagine creata nel passo [“3”](#) a pagina 485.
6. Eseguire la funzione LOAD di CSQUTIL per caricare nuovamente tutti i messaggi salvati durante il passo [“2”](#) a pagina 485.
7. Consentire a tutti gli utenti di accedere al gestore code.
8. Eliminare la vecchia serie di pagine di grandi dimensioni.

## Come reintrodurre una serie di pagine

In alcuni scenari è utile riportare una vecchia serie di pagine in linea al gestore code. A meno che non venga eseguita un'azione specifica, quando la vecchia serie di pagine viene portata in linea, il gestore code riconoscerà che l'RBA di recupero della serie di pagine memorizzato nella serie di pagine stessa e nei record del punto di controllo è obsoleto e avvierà automaticamente il ripristino del supporto della serie di pagine per aggiornarlo.

Tale ripristino del supporto può essere eseguito solo al riavvio del gestore code ed è probabile che impiega un periodo di tempo considerevole, soprattutto se è necessario leggere i log di archivio presenti sul nastro. Tuttavia, normalmente in questa circostanza, la serie di pagine è stata offline per il periodo di tempo e quindi il log non contiene informazioni relative al ripristino della serie di pagine.

Sono disponibili le seguenti tre scelte:

**Consentire l'esecuzione del ripristino completo del supporto.**

1. Chiudere il gestore code.
2. Verificare che le definizioni siano disponibili per la serie di pagine sia nella procedura di attività avviata per il gestore code che nel dataset di inizializzazione CSQINP1 .
3. Riavviare il gestore code.

**Consentire l'eliminazione di tutti i messaggi sulla serie di pagine.**

Questa scelta è utile quando un set di pagine è stato offline per molto tempo (alcuni mesi, ad esempio) ed è stato deciso di riutilizzarlo per uno scopo diverso.

1. Formattare la serie di pagine utilizzando la funzione FORMAT di CSQUTIL con l'opzione TYPE (NEW).
2. Aggiungere definizioni per la serie di pagine sia alla procedura dell'attività avviata per il gestore code che al dataset di inizializzazione CSQINP1 .
3. Riavviare il gestore code.

L'utilizzo dell'opzione TYPE (NEW) per la formattazione cancella il contenuto corrente della serie di pagine e indica al gestore code di ignorare qualsiasi informazione cronologica nel punto di controllo relativa alla serie di pagine.

#### **Portare la serie di pagine in linea evitando il processo di ripristino dei supporti.**

Utilizzare questa tecnica solo se si è certi che la serie di pagine è stata offline da un arresto pulito del gestore code. Questa scelta è più appropriata quando la serie di pagine è stata offline per un breve periodo, in genere a causa di problemi operativi come l'esecuzione di un backup durante l'avvio del gestore code.

1. Formattare la serie di pagine utilizzando la funzione FORMAT di CSQUTIL con l'opzione TYPE (REPLACE).
2. Aggiungere di nuovo la serie di pagine nel gestore code in modo dinamico utilizzando il comando DEFINE PSID con l'opzione DSN o consentirne l'aggiunta al riavvio del gestore code.

L'utilizzo dell'opzione TYPE (REPLACE) per la formattazione verifica che la serie di pagine sia stata chiusa correttamente dal gestore code e la contrassegna in modo che non venga eseguito il ripristino del supporto. Non vengono apportate altre modifiche al contenuto della serie di pagine.

### **Come eseguire il backup e il ripristino delle serie di pagine**

Esistono diversi meccanismi disponibili per il backup e il recupero. Utilizzare questo argomento per comprendere questi meccanismi.

Questa sezione descrive i seguenti argomenti:

- [“Creazione di un punto di recupero per le risorse non condivise” a pagina 486](#)
- [“Backup delle serie di pagine” a pagina 487](#)
- [“Ripristino delle serie di pagine” a pagina 488](#)
- [Come eliminare le serie di pagine](#)

Per informazioni su come creare un punto di ripristino per le risorse condivise, consultare [“Ripristino delle code condivise” a pagina 495](#).

### **Creazione di un punto di recupero per le risorse non condivise**

IBM MQ può ripristinare gli oggetti e i messaggi persistenti non condivisi allo stato corrente se:

1. Esistono copie delle serie di pagine da un punto precedente.
2. Tutti i log IBM MQ sono disponibili per eseguire il ripristino da quel punto.

Questi rappresentano un punto di recupero per le risorse non condivise.

Sia gli oggetti che i messaggi vengono conservati nelle serie di pagine. Sulla stessa serie di pagine possono esistere più oggetti e messaggi da code differenti. Per scopi di ripristino, non è possibile eseguire il backup di oggetti e messaggi in isolamento, pertanto è necessario eseguire il backup di una serie di pagine per garantire il corretto ripristino dei dati.

Il log di ripristino di IBM MQ contiene un record di tutti i messaggi persistenti e le modifiche apportate agli oggetti. Se IBM MQ non riesce (ad esempio, a causa di un errore I/O su una serie di pagine), è possibile ripristinare la serie di pagine ripristinando la copia di backup e riavviando il gestore code. IBM MQ applica le modifiche di log alla serie di pagine dal punto della copia di backup.

Esistono due modi per creare un punto di ripristino:

### **Backup completo**

Arrestare il gestore code, che forza tutti gli aggiornamenti alle serie di pagine.

Ciò consente di riavviare dal punto di recupero, utilizzando solo i dataset della serie di pagine di cui è stato eseguito il backup e i log da quel punto in poi.

### **Backup inesatto**

Eseguire copie di backup *inesatte* delle serie di pagine senza arrestare il gestore code.

Se si utilizza questo metodo e i log associati successivamente vengono danneggiati o persi, non è possibile utilizzare le copie di backup della serie di pagine inesatte per il ripristino. Ciò è dovuto al fatto che le copie di backup della serie di pagine non esatte contengono una vista incongruente dello stato del gestore code e dipendono dai log disponibili. Se i log non sono disponibili, è necessario tornare all'ultima serie di copie della serie di pagine di backup eseguite mentre il sottosistema era inattivo ( [Metodo 1](#) ) e accettare la perdita di dati da quel momento.

### **Metodo 1: backup completo**

Questo metodo implica la chiusura del gestore code. Ciò forza tutti gli aggiornamenti alle serie di pagine in modo che le serie di pagine siano in uno stato congruente.

1. Arrestare tutte le applicazioni IBM MQ che utilizzano il gestore code (consentendone il primo completamento). Ciò può essere fatto modificando la sicurezza di accesso o le impostazioni della coda, ad esempio.
2. Quando tutte le attività sono state completate, visualizzare e risolvere eventuali unità di ripristino in dubbio. Utilizzare i comandi DISPLAY CONN e RESOLVE INDOUBT, come descritto in [DISPLAY CONN](#) e [RESOLVE INDOUBT](#).

Ciò porta le serie di pagine ad uno stato congruente; se non si esegue questa operazione, le serie di pagine potrebbero non essere congruenti e si sta effettivamente eseguendo un backup inesatto.

3. Immettere il comando ARCHIVE LOG per garantire che i dati di log più recenti vengano scritti nei dataset di log.
4. Immettere il comando STOP QMGR MODE (QUIESCE). Registrare il valore RBA più basso nei messaggi CSQI024I o CSQI025I (per ulteriori informazioni, consultare [CSQI024I](#) e [CSQI025I](#) ). È necessario mantenere i dataset di registrazione a partire da quello indicato dal valore RBA fino al dataset di registrazione corrente.
5. Eseguire il backup di tutte le serie di pagine del gestore code (consultare [“Backup delle serie di pagine”](#) a pagina 487 ).

### **Metodo 2: backup fuzzy**

Questo metodo non implica la chiusura del gestore code. Pertanto, gli aggiornamenti potrebbero trovarsi nei buffer di memoria virtuale durante il processo di backup. Ciò significa che le serie di pagine non si trovano in uno stato congruente e possono essere utilizzate solo per il recupero con i log.

1. Emettere il comando DISPLAY USAGE TYPE (ALL) e registrare il valore RBA nei messaggi CSQI024I o CSQI025I (consultare [CSQI024I](#) e [CSQI025I](#) per ulteriori informazioni).
2. Eseguire il backup delle serie di pagine (consultare [“Backup delle serie di pagine”](#) a pagina 487 ).
3. Immettere il comando ARCHIVE LOG per assicurarsi che i dati di log più recenti siano scritti nei dataset di log. Per riavviare dal punto di ripristino, è necessario mantenere i dataset di log a partire dal dataset di log indicato dal valore RBA fino al dataset di log corrente.

## **Backup delle serie di pagine**

Per recuperare una serie di pagine, IBM MQ deve sapere fino a che punto si trova nel log. IBM MQ conserva un numero RBA di log nella pagina zero di ogni serie di pagine, denominato LSN ( *recovery log sequence number* ). Questo numero è l'RBA iniziale nel log da cui IBM MQ può recuperare la serie di pagine. Quando si esegue il backup di una serie di pagine, viene copiato anche questo numero.

Se la copia viene utilizzata successivamente per recuperare la serie di pagine, IBM MQ deve avere accesso a tutti i record di log da questo valore RBA all'RBA corrente. Ciò significa che è necessario conservare una quantità sufficiente di record di log per consentire a IBM MQ di eseguire il ripristino dalla copia di backup più vecchia di una serie di pagine che si intende conservare.

Utilizzare la funzione ADRDSSU COPY per copiare le serie di pagine.

Per ulteriori informazioni, consultare la documentazione [COPY DATASET Command Syntax for Logical Data Set](#).

Ad esempio:

```
//STEP2 EXEC PGM=ADRDSSU,REGION=6M
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
COPY -
DATASET(INCLUDE(SCENDATA.MQPA.PAGESET.*)) -
RENAMEU(SCENDATA.MQPA.PAGESET.** ,SCENDATA.MQPA.BACKUP1.** ) -
SPHERE -
REPUNC -
FASTREPLICATION(PREF ) -
CANCELERROR -
TOL(ENQF)
/*
//
```

Se si copia la serie di pagine mentre il gestore code è in esecuzione, è necessario utilizzare un programma di utilità di copia che copia prima la pagina zero della serie di pagine. Se non si esegue questa operazione, è possibile corrompere i dati nella serie di pagine.

Se il processo di espansione dinamica di una serie di pagine viene interrotto, ad esempio a causa della perdita di alimentazione del sistema, è ancora possibile utilizzare ADRDSSU per eseguire una copia di riserva di una serie di pagine.

Se si esegue un IDCAMS (Access Method Services) LISTCAT ENT('page set data set name') ALLOC, si vedrà che HI-ALLOC-RBA è superiore a HI-USED-RBA.

La prossima volta che questa serie di pagine si riempie viene nuovamente estesa, se possibile, e vengono utilizzate le pagine tra l'RBA più utilizzato e l'RBA più elevato assegnato, insieme a un'altra nuova estensione.

## Backup delle definizioni oggetto

È inoltre necessario eseguire il backup di copie delle proprie definizioni di oggetto. Per fare ciò, utilizzare la funzione MAKEDEF della funzione CSQUTIL COMMAND (descritta in [Immissione di comandi in IBM MQ \(COMMAND\)](#)).

Eseguire il backup delle definizioni degli oggetti ogni volta che si esegue una copia di backup del gestore code e si conserva la versione più aggiornata.

## Ripristino delle serie di pagine

Se il gestore code è stato terminato a causa di un errore, il gestore code può essere normalmente riavviato con tutto il ripristino eseguito durante il riavvio. Tuttavia, tale ripristino non è possibile se non sono disponibili serie di pagine o serie di dati di log. La misura in cui è ora possibile eseguire il ripristino dipende dalla disponibilità delle copie di backup delle serie di pagine e delle serie di dati di log.

Per riavviare da un punto di ripristino, è necessario avere:

- Una copia di backup della serie di pagine da recuperare.
- Se è stato utilizzato il processo di backup "fuzzy" descritto in ["Metodo 2: backup fuzzy"](#) a pagina 487, il dataset di log che includeva il valore RBA registrato, il dataset di log creato dal comando ARCHIVE LOG e tutti i dataset di log tra questi.

- Se è stato utilizzato il backup completo, ma non si dispone dei dataset di log successivi a quelli eseguiti dal comando ARCHIVE LOG, **non** è necessario eseguire la funzione FORMAT TYPE (REPLACE) del programma di utilità CSQUTIL su tutte le serie di pagine.

Per ripristinare una pagina impostata allo stato corrente, è necessario disporre anche di tutti i record e i dataset di log dal comando ARCHIVE LOG.

Esistono due metodi per ripristinare una serie di pagine. Per utilizzare uno dei due metodi, è necessario arrestare il gestore code.

### **Ripristino semplice**

Questo è il metodo più semplice ed è appropriato per la maggior parte delle situazioni di ripristino.

1. Eliminare la serie di pagine che si desidera ripristinare dal backup.
2. Utilizzare la funzione ADRDSSU COPY per recuperare la serie di pagine dalla copia di backup.

In alternativa, è possibile ridenominare la copia di backup con il nome originale oppure modificare l'istruzione DD CSQP00xx nella procedura del gestore code in modo che faccia riferimento alla serie di pagine di backup. Tuttavia, se si perde o si danneggia la serie di pagine, non si avrà più una copia di backup da cui ripristinare.

3. Riavviare il gestore code.
4. Una volta riavviato correttamente il gestore code, è possibile riavviare le applicazioni
5. Ripristinare le normali procedure di backup per la pagina ripristinata.

### **Ripristino avanzato**

Questo metodo fornisce vantaggi in termini di prestazioni se si dispone di una serie di pagine di grandi dimensioni da ripristinare o se è stata eseguita molta attività sulla serie di pagine da quando è stata eseguita l'ultima copia di backup. Tuttavia, richiede un intervento manuale superiore rispetto al metodo semplice, che potrebbe aumentare il rischio di errori e il tempo impiegato per eseguire il recupero.

1. Eliminare e ridefinire la serie di pagine che si desidera ripristinare dal backup.
2. Utilizzare ADRDSSU per copiare la copia di backup della serie di pagine nella nuova serie di pagine. Definire la nuova serie di pagine con un valore di estensione secondario in modo che possa essere espansa dinamicamente.

In alternativa, è possibile ridenominare la copia di backup con il nome originale oppure modificare l'istruzione DD CSQP00xx nella procedura del gestore code in modo che faccia riferimento alla serie di pagine di backup. Tuttavia, se si perde o si danneggia la serie di pagine, non si avrà più una copia di backup da cui ripristinare.

3. Modificare le definizioni CSQINP1 per il gestore code per rendere il pool di buffer associato alla serie di pagine ripristinata il più grande possibile. Rendendo il pool di buffer grande, è possibile mantenere tutte le pagine modificate residenti nel pool di buffer e ridurre la quantità di I/O nella serie di pagine.
4. Riavviare il gestore code.
5. Una volta riavviato correttamente il gestore code, arrestarlo (utilizzando il quiesce) e riavviarlo utilizzando la normale definizione del pool di buffer per tale serie di pagine. Una volta completato correttamente questo secondo riavvio, è possibile riavviare le applicazioni
6. Ripristinare le normali procedure di backup per la pagina ripristinata.

### **Cosa accade quando il gestore code viene riavviato**

Quando il gestore code viene riavviato, applica tutte le modifiche apportate alla serie di pagine registrate nel log, a partire dal punto di riavvio della serie di pagine. IBM MQ può ripristinare più serie di pagine in questo modo. La serie di pagine viene espansa dinamicamente, se necessario, durante il ripristino del supporto.

Durante il riavvio, IBM MQ determina l'RBA di log da cui iniziare prendendo il valore più basso da quanto segue:

- LSN di ripristino dal record di log del checkpoint per ogni serie di pagine.
- LSN di ripristino dalla pagina zero in ogni serie di pagine.
- L'RBA dell'unità di ripristino incompleta più vecchia nel sistema nel momento in cui è stato eseguito il backup.

Tutte le definizioni oggetto vengono memorizzate nella serie di pagine zero. I messaggi possono essere memorizzati su qualsiasi serie di pagine disponibile.

**Nota:** Il gestore code non può essere riavviato se la serie di pagine zero non è disponibile.

## Come eliminare le serie di pagine

Si elimina una serie di pagine utilizzando il comando DELETE PSID; per dettagli su questo comando, consultare [DELETE PSID](#) .

Non è possibile eliminare una serie di pagine a cui fa ancora riferimento una classe di memoria. Utilizzare DISPLAY STGCLASS per individuare quali classi di memoria fanno riferimento a una serie di pagine.

L'assegnazione del dataset è stata annullata da IBM MQ ma non è stata eliminata. Rimane disponibile per un utilizzo futuro o può essere eliminato utilizzando le funzioni z/OS .

Rimuovere la serie di pagine dalla procedura di attività avviata per il gestore code.

Rimuovere la definizione della serie di pagine dal dataset di inizializzazione CSQINP1 .

## Come eseguire il backup e ripristinare le code utilizzando CSQUTIL

Utilizzare questo argomento come riferimento per ulteriori informazioni sul backup e sul ripristino mediante CSQUTIL.

È possibile utilizzare le funzioni di utilità CSQUTIL per eseguire il backup e il ripristino delle code. Per eseguire il backup di una coda, utilizzare la funzione COPY o SCOPY per copiare i messaggi da una coda in un dataset. Per ripristinare la coda, utilizzare la funzione complementare LOAD o SLOAD. Per ulteriori informazioni, consultare [Programma di utilità IBM MQ](#).

## Gestione dei pool di buffer

Utilizzare questo argomento se si desidera modificare o eliminare i pool di buffer.

Questo argomento descrive come modificare ed eliminare i pool di buffer. Contiene le seguenti sezioni:

- [“Come modificare il numero di buffer in un pool di buffer” a pagina 491](#)
- [“Come eliminare un pool di buffer” a pagina 491](#)

I pool di buffer vengono definiti durante l'inizializzazione del gestore code, utilizzando comandi [DEFINE BUFFPOOL](#) emessi dal dataset di input di inizializzazione CSQINP1. I loro attributi possono essere modificati in risposta ai requisiti di business mentre il gestore code è in esecuzione, utilizzando i processi descritti in questo argomento. Il gestore code registra gli attributi del pool di buffer correnti nei record di log del checkpoint. Questi vengono ripristinati automaticamente al successivo riavvio del gestore code, a meno che la definizione del pool di buffer in CSQINP1 non includa l'attributo REPLACE.

Utilizzare il comando [DISPLAY USAGE](#) per visualizzare gli attributi del buffer correnti.

È anche possibile definire i pool di buffer in maniera dinamica utilizzando il comando [DEFINE PSID](#) con l'opzione DSN.

Se si modificano i pool di buffer in modo dinamico, è necessario aggiornare anche le relative definizioni nel dataset di inizializzazione CSQINP1.

Consultare Pianificazione dei messaggi z/OS per una descrizione delle serie di pagine, delle classi di memoria, dei buffer e dei pool di buffer e per alcune delle considerazioni sulle prestazioni applicabili.

**Nota:** I pool di buffer utilizzano una memoria significativa. Quando si aumenta la dimensione di un pool di buffer o si definisce un nuovo pool di buffer, assicurarsi che sia disponibile memoria sufficiente. Per ulteriori informazioni, consultare [Memoria spazio di indirizzo](#).

## Come modificare il numero di buffer in un pool di buffer

Se un pool di buffer è troppo piccolo, la condizione può risultare nel messaggio [CSQP020E](#) sulla console, è possibile assegnarvi più buffer utilizzando il comando ALTER BUFFPOOL nel modo seguente:

1. Determinare quanto spazio è disponibile per i nuovi buffer esaminando i messaggi [CSQY220I](#) nel log. Lo spazio disponibile è riportato in MB. Poiché un buffer ha una dimensione di 4 KB, ogni MB di spazio disponibile consente di allocare 256 buffer. Non assegnare tutto lo spazio libero ai buffer, poiché alcuni sono necessari per altre attività.

Se il pool di buffer utilizza pagine da 4 KB fisse, ovvero, il relativo attributo PAGECLAS è FIXED4KB, assicurarsi che vi sia sufficiente memoria reale disponibile sulla LPAR.

2. Se lo spazio libero riportato non è sufficiente, rilasciare alcuni buffer da un altro pool di buffer utilizzando il comando

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

dove *buf - pool - id* è il pool di buffer da cui si desidera recuperare spazio e *integer* è il nuovo numero di buffer da assegnare a questo pool di buffer, che deve essere inferiore al numero originale di buffer ad esso assegnati.

3. Aggiungere i buffer al pool di buffer che si desidera espandere utilizzando il comando

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

dove *buf - pool - id* è il pool di buffer da espandere e *integer* è il nuovo numero di buffer da assegnare a questo pool di buffer, che deve essere maggiore del numero originale di buffer ad esso assegnati.

## Come eliminare un pool di buffer

Quando un pool di buffer non viene più utilizzato da alcuna serie di pagine, eliminarlo per rilasciare la memoria virtuale ad esso assegnata.

Eliminare un pool di buffer utilizzando il comando [DELETE BUFFPOOL](#) . Il comando ha esito negativo se una serie di pagine utilizza questo pool di buffer.

Consultare [“Come eliminare le serie di pagine”](#) a pagina 490 per informazioni su come eliminare le serie di pagine.

## **Gestione di gruppi di condivisione code e code condivise su z/OS**

IBM MQ può utilizzare diversi tipi di risorse condivise, ad esempio gruppi di condivisione code, code condivise e CF (Coupling Facility). Utilizzare questo argomento per esaminare le procedure necessarie per gestire queste risorse condivise.

Questa sezione contiene informazioni sui seguenti argomenti:

- [“Gestione dei gruppi di condivisione code”](#) a pagina 492
- [“Gestione delle code condivise”](#) a pagina 495
- [“Gestione degli oggetti del gruppo”](#) a pagina 500
- [“Gestione della CF \(coupling facility\)”](#) a pagina 500

## **Gestione dei gruppi di condivisione code**

È possibile aggiungere o rimuovere un gestore code a un gruppo di condivisione code (QSG) e gestire le tabelle Db2 associate.

Questo argomento contiene sezioni relative alle seguenti attività:

- [“Impostazione di un gruppo di condivisione code” a pagina 492](#)
- [“Aggiunta di un gestore code ad un gruppo di condivisione code” a pagina 493](#)
- [“Rimozione di un gestore code da un gruppo di condivisione code” a pagina 493](#)
- [“Rimozione di un gruppo di condivisione code dalle tabelle Db2” a pagina 494](#)
- [“Convalida della coerenza delle definizioni Db2” a pagina 495](#)

### **Impostazione di un gruppo di condivisione code**

Ciascun gruppo di condivisione code ha un nome composto da un massimo di quattro caratteri. Il nome deve essere univoco nella rete e diverso da qualsiasi altro nome di gestore code.

Seguire questa procedura per impostare un gruppo di condivisione code:

1. Se questo è il primo gruppo di condivisione code ad utilizzare il gruppo di condivisione dati Db2 , configurare l'ambiente Db2.
2. [Impostare la CF \(coupling facility\).](#)
3. Aggiungere il gruppo di condivisione code alle tabelle Db2 . Utilizzare la funzione ADD QSG del programma di utilità del gruppo di condivisione code (CSQ5PQSG). Questo programma è descritto in [Il programma di utilità del gruppo di condivisione code](#). Un esempio viene fornito in `thlqual.SCSQPROC(CSQ45AQS)`.
4. Aggiungere un gestore code al gruppo di condivisione code effettuando le operazioni riportate in [“Aggiunta di un gestore code ad un gruppo di condivisione code” a pagina 493](#) .
5. Definisci le strutture dell'applicazione in IBM MQ seguendo la procedura in [“Aggiunta di una struttura CFS \(coupling facility structure\)” a pagina 500](#).
6. Se necessario, [migrare le code non condivise nelle code condivise](#).
7. Per la disponibilità, creare canali condivisi all'interno e all'esterno del gruppo di condivisione code.
  - Per le connessioni nel gruppo di condivisione code:
    - Configurare un socket VIPA o un router hardware per distribuire il carico di lavoro tra i gestori code disponibili in QSG.
    - Definire un canale ricevente con QSGDISP (GROUP), per garantire che la definizione del canale sia disponibile su tutti i gestori code in QSG.
    - Avviare un listener con INDISP (GROUP), su ogni gestore code, per connessioni del canale MCA nel QSG. Le connessioni client nel QSG devono ancora connettersi a un listener avviato con INDISP (QMGR).
    - Modificare le applicazioni per connettersi utilizzando il nome QSG, piuttosto che un nome gestore code specifico.
    - Assicurarsi che le regole di autenticazione di canale su tutti i gestori code in QSG siano le stesse, per consentire alle applicazioni di connettersi a qualsiasi gestore code in QSG.
  - Per le connessioni al di fuori del gruppo di condivisione code:
    - Definire una coda di trasmissione condivisa.
    - Definire il canale in uscita con QSGDISP (GROUP) e DEFCDISP (SHARED).

Se si converte un canale esistente in un canale condiviso, potrebbe essere necessario immettere il comando `RESET CHANNEL` prima di avviare il canale in quanto la coda di sincronizzazione utilizzata dal canale sarà stata modificata.

## Aggiunta di un gestore code ad un gruppo di condivisione code

Un gestore code può essere aggiunto a un gruppo di condivisione code esistente.

Si noti quanto segue:

- Il gruppo di condivisione code deve esistere prima di poter aggiungere i gestori code.
- Un gestore code può essere membro di un solo gruppo di condivisione code.

Per aggiungere un gestore code a un gruppo di condivisione code, procedere come segue:

1. Eseguire le attività in implementare i controlli di sicurezza ESM per il gruppo di condivisione code per concedere l'accesso appropriato agli ID utente del gestore code e dell'iniziatore di canali.
2. Se il gruppo di condivisione code ha strutture CF configurate per scaricare i dati in SMDS, eseguire le attività in impostare l'ambiente SMDS.
3. Chiudere il gestore code.
4. Utilizzare la funzione ADD QMGR del programma di utilità del gruppo di condivisione code (CSQ5PQSG). Questo programma è descritto in il programma di utilità del gruppo di condivisione code. Un esempio è fornito in `thlqual.SCSQPROC(CSQ45AQM)`.
5. Modificare il modulo dei parametri di sistema per aggiungere i dati del gruppo di condivisione code:
  - a. Modificare CSQ6SYSP per specificare il parametro QSGDATA. Per ulteriori informazioni, consultare Utilizzo di CSQ6SYSP.
  - b. Assemblare e collegare il modulo dei parametri di sistema. È possibile utilizzare un nome diverso per il modulo di caricamento.
  - c. Modificare il processo di avvio per utilizzare il nuovo modulo.
6. Copiare e personalizzare il membro di esempio `thlqual.SCSQPROC(CSQ4INSS)`, che definisce le strutture CF e le code SYSTEM richieste. Aggiungere il membro personalizzato alla DD CSQINP2 nel JCL di avvio del gestore code.
7. Riavviare il gestore code utilizzando il modulo del parametro di sistema del gruppo di condivisione code.
8. Facoltativamente, eseguire la migrazione ai profili di sicurezza prefissati dal nome del gruppo di condivisione code invece che dal nome del gestore code.
9. Se i canali condivisi vengono utilizzati per le connessioni in QSG, creare regole di autenticazione del canale che riflettano quelle sugli altri gestori code in QSG, per consentire alle applicazioni di connettersi a qualsiasi gestore code in QSG.
10. 10. Facoltativamente, effettuare una delle seguenti operazioni per consentire alle applicazioni connesse al gestore code in QSG di inserire i messaggi nelle code ospitate da altri gestori code in QSG:
  - Attivare l' accodamento all'interno del gruppo immettendo il comando `ALTER QMGR IGQ (ENABLED)`.
  - Definire le code di trasmissione e i canali per gli altri gestori code in QSG. La definizione di code di trasmissione con lo stesso nome dei gestori code di destinazione evita la necessità di definire code remote e alias dei gestori code.

**Nota:** Per aggiungere un gestore code a un gruppo di condivisione code esistente contenente gestori code che eseguono versioni precedenti di IBM MQ, è necessario prima applicare la PTF di coesistenza per la versione più alta di IBM MQ nel gruppo ad ogni gestore code della versione precedente del gruppo.

## Rimozione di un gestore code da un gruppo di condivisione code

È possibile rimuovere un gestore code da un gruppo di condivisione code solo se i log del gestore code non sono necessari da un altro processo e tutti gli SMDS di proprietà del gestore code sono vuoti.

Per ulteriori informazioni, consultare Eliminazione dei dataset di messaggi condivisi e DELETE CFSTRUCT.

I log sono necessari se contengono:

- L'ultimo backup di una delle strutture dell'applicazione CF (Coupling Facility) utilizzata dal gruppo di condivisione code
- I dati necessari per un processo di ripristino futuro, ovvero, il gestore code ha utilizzato una struttura recuperabile dall'ora descritta dall'ultimo valore dell'intervallo di esclusione di backup.

Se si applica uno o entrambi questi punti o se un SMDS di proprietà del gestore code contiene messaggi, il gestore code non può essere rimosso. Per determinare quali log dei gestori code sono necessari per un processo di ripristino futuro, utilizzare il comando MQSC DISPLAY CFSTATUS con l'opzione TYPE (BACKUP) (per i dettagli di questo comando, vedere [DISPLAY CFSTATUS](#) ).

Utilizzare la seguente procedura per rimuovere un gestore code da un gruppo di condivisione code:

1. Arrestare tutte le applicazioni connesse al gestore code che hanno inserito i messaggi nelle code condivise.
2. Risolvere eventuali unità di lavoro dubbie che coinvolgono questo gestore code.
3. Determinare se sono presenti messaggi in qualsiasi SMDS di proprietà del gestore code immettendo il comando DISPLAY USAGE TYPE (SMDS).
4. Se esistono messaggi scaricati per qualsiasi struttura dell'applicazione, attendere che tali messaggi siano stati richiamati dalla coda. Il numero di messaggi con offload riportati da SMDS (DISPLAY USAGE TYPE) deve essere zero prima di continuare.
5. Chiudere il gestore code utilizzando STOP QMGR MODE (QUIESCE).
6. Attendere un intervallo almeno equivalente al valore del parametro EXCLINT che verrà specificato nel comando BACKUP CFSTRUCT nel passo successivo.
7. Su un altro gestore code, eseguire un backup della struttura CF per ogni struttura CF ripristinabile utilizzando il comando MQSC BACKUP CFSTRUCT e specificando un valore EXCLINT come richiesto nel passo precedente.
8. Confermare che i log del gestore code non sono necessari per ripristinare le strutture CF, controllando l'output del comando DISPLAY CFSTATUS (\*) TYPE (BACKUP).
9. Utilizzare la funzione REMOVE QMGR del programma di utilità CSQ5PQSG per eliminare il gestore code dal gruppo di condivisione code. Questo programma è descritto in [Il programma di utilità del gruppo di condivisione code](#). Un esempio viene fornito in thlqual.SCSQPROC(CSQ45RQM).
10. Prima di riavviare il gestore code, reimpostare il valore predefinito del parametro di sistema QSGDATA e ricrearlo. Consultare [Utilizzo di CSQ6SYSP](#) per informazioni su come adattare i parametri di sistema.

Tenere presente che quando si rimuove l'ultimo gestore code in un gruppo di condivisione code, è necessario utilizzare l'opzione FORCE anziché REMOVE. Ciò rimuove il gestore code dal gruppo di condivisione code, senza eseguire i controlli di congruenza dei log del gestore code richiesti per il recupero. Eseguire questa operazione solo se si sta eliminando il gruppo di condivisione code.

## Rimozione di un gruppo di condivisione code dalle tabelle Db2

Per rimuovere un gruppo di condivisione code dalle tabelle Db2 , utilizzare la funzione REMOVE QSG del programma di utilità del gruppo di condivisione code (CSQ5PQSG). Questo programma è descritto in [Il programma di utilità del gruppo di condivisione code](#). Un esempio viene fornito in thlqual.SCSQPROC(CSQ45RQS).

È possibile rimuovere un gruppo di condivisione code solo dalle tabelle del gruppo di condivisione dati Db2 comuni dopo aver rimosso tutti i gestori code dal gruppo di condivisione code (come descritto in ["Rimozione di un gestore code da un gruppo di condivisione code"](#) a pagina 493 ).

Quando il record del gruppo di condivisione code viene eliminato dalla tabella di amministrazione del gruppo di condivisione code, tutti gli oggetti e le informazioni di gestione relativi a tale gruppo di condivisione code vengono eliminati da altre tabelle IBM MQ Db2 . Ciò include le informazioni sulla coda condivisa e sull'oggetto gruppo.

## Convalida della coerenza delle definizioni Db2

I problemi per le code condivise all'interno di un gruppo di condivisione code possono verificarsi se le definizioni dell'oggetto Db2, per qualsiasi motivo, sono diventate incongruenti.

Per convalidare la congruenza delle definizioni degli oggetti Db2 per i gestori code, le strutture CF e le code condivise, utilizzare la funzione VERIFY QSG del programma di utilità del gruppo di condivisione code (CSQ5PQSG). Questo programma è descritto in [Il programma di utilità del gruppo di condivisione code](#).

### **Gestione delle code condivise**

Utilizzare questo argomento per comprendere come ripristinare, spostare e migrare le code condivise.

Questa sezione descrive le seguenti attività:

- [“Ripristino delle code condivise” a pagina 495](#)
- [“Spostamento delle code condivise” a pagina 496](#)
- [“Migrazione di code non condivise in code condivise” a pagina 498](#)
- [Sospensione di una connessione Db2](#)

## Ripristino delle code condivise

IBM MQ può ripristinare i messaggi persistenti sulle code condivise se:

- I backup delle strutture CF che contengono i messaggi sono stati eseguiti.
- Tutti i log per tutti i gestori code nel gruppo di condivisione code sono disponibili, per eseguire il recupero dal momento in cui vengono eseguiti i backup.
- Db2 è disponibile e la tabella di backup della struttura è più recente del backup della struttura CF più recente.

I messaggi su una coda condivisa sono memorizzati in una struttura CF (coupling facility). I messaggi persistenti possono essere inseriti nelle code condivise e, come i messaggi persistenti nelle code non condivise, vengono copiati nel log del gestore code. I comandi MQSC [BACKUP CFSTRUCT](#) e [RECOVER CFSTRUCT](#) vengono forniti per consentire il recupero di una struttura CF nel caso improbabile di un malfunzionamento della CF. In tali circostanze, tutti i messaggi non persistenti memorizzati nella struttura interessata vengono persi, ma i messaggi persistenti possono essere recuperati. Qualsiasi ulteriore attività dell'applicazione che utilizza la struttura viene impedita fino a quando la struttura non viene ripristinata.

Per abilitare il ripristino, è necessario eseguire frequentemente il backup delle strutture dell'elenco CF utilizzando il comando MQSC [BACKUP CFSTRUCT](#). I messaggi nella struttura CF vengono scritti nel dataset del log attivo del gestore code che esegue il backup. Scrive un record del backup in Db2: il nome della struttura CF di cui si sta eseguendo il backup, il nome del gestore code che esegue il backup, l'intervallo RBA per questo backup sul log del gestore code e l'ora di backup. Eseguire il backup delle strutture dell'elenco CF anche se non si utilizzano attivamente le code condivise, ad esempio, se è stato impostato un gruppo di condivisione code che intende utilizzarlo in futuro.

È possibile ripristinare una struttura CF emettendo un comando MQSC [RECOVER CFSTRUCT](#) al gestore code che può eseguire il recupero; è possibile utilizzare qualsiasi gestore code nel gruppo di condivisione code. È possibile specificare una singola struttura CF da recuperare oppure è possibile recuperare più strutture CF contemporaneamente.

Come notato in precedenza, è importante eseguire il backup delle strutture dell'elenco CF frequentemente, altrimenti il recupero di una struttura CF può richiedere molto tempo. Inoltre, il processo di recupero non può essere annullato.

La definizione di una coda condivisa viene conservata in un database Db2 e può quindi essere ripristinata, se necessario, utilizzando le procedure standard del database Db2. Per ulteriori informazioni, consultare [Code condivise e gruppi di condivisione code](#).

## Spostamento delle code condivise

Questa sezione descrive come eseguire il bilanciamento del carico spostando una coda condivisa da una struttura CFS ad un'altra. Descrive inoltre come spostare una coda non condivisa in una coda condivisa e come spostare una coda condivisa in una coda non condivisa.

Quando si sposta una coda, è necessario definire una coda temporanea come parte della procedura. Questo perché ogni coda deve avere un nome univoco, quindi non è possibile avere due code con lo stesso nome, anche se le code hanno disposizioni di coda differenti. IBM MQ tollera di avere due code con lo stesso nome (come nel passo [“2”](#) a pagina 496 ), ma non è possibile utilizzare le code.

- Spostamento di una coda da una struttura CFS a un'altra
- Spostamento di una coda non condivisa in una coda condivisa
- Spostamento di una coda condivisa in una coda non condivisa

### Spostamento di una coda da una struttura CFS a un'altra

Per spostare le code ed i relativi messaggi da una struttura CF ad un'altra, utilizzare il comando MQSC `MOVE QLOCAL` . Una volta identificate la coda o le code che si desidera spostare in una nuova struttura CF, utilizzare la seguente procedura per spostare ciascuna coda:

1. Verificare che la coda che si desidera spostare non venga utilizzata da alcuna applicazione, ovvero che gli attributi della coda IPPROCS e OPPOCS siano zero su tutti i gestori code nel gruppo di condivisione code.
2. Impedire alle applicazioni di inserire i messaggi nella coda spostati modificando la definizione della coda per disabilitare MQPUT . Modificare la definizione della coda in PUT (DISABLED).
3. Definire una coda temporanea con gli stessi attributi della coda spostata utilizzando il seguente comando:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

**Nota:** Se questa coda temporanea esiste da una precedente esecuzione, eliminarla prima di eseguire la definizione.

4. Spostare i messaggi nella coda temporanea utilizzando il seguente comando:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Eliminare la coda che si sta spostando, utilizzando il comando:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Ridefinire la coda che si sta spostando, modificando l'attributo CFSTRUCT, utilizzando il seguente comando:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
```

Quando la coda viene ridefinita, si basa sulla coda temporanea creata nella fase [“3”](#) a pagina 496.

7. Spostare nuovamente i messaggi nella nuova coda utilizzando il comando:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

8. La coda creata nel passo “3” a pagina 496 non è più richiesta. Utilizzare il seguente comando per eliminarla:

```
DELETE QL(TEMP_QUEUE)
```

9. Se la coda da spostare è stata definita nei dataset CSQINP2 , modificare l'attributo CFSTRUCT del comando DEFINE QLOCAL appropriato nei dataset CSQINP2 . Aggiungere la parola chiave REPLACE in modo che la definizione della coda esistente venga sostituita.

Figura 38 a pagina 497 mostra un lavoro campione per lo spostamento di una coda da una struttura CF ad un'altra.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*
```

Figura 38. Lavoro di esempio per lo spostamento di una coda da una struttura CF a un'altra

### Spostamento di una coda non condivisa in una coda condivisa

La procedura per spostare una coda non condivisa in una coda condivisa è simile alla procedura per spostare una coda da una struttura CF a un'altra (consultare “Spostamento di una coda da una struttura CFS a un'altra” a pagina 496 ). Figura 39 a pagina 498 fornisce un lavoro di esempio per eseguire questa operazione.

**Nota:** Tenere presente che i messaggi sulle code condivise sono soggetti a determinate limitazioni sulla dimensione massima dei messaggi, sulla persistenza dei messaggi e sul tipo di indice della coda, pertanto potrebbe non essere possibile spostare alcune code non condivise in una coda condivisa.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*

```

*Figura 39. Lavoro di esempio per lo spostamento di una coda non condivisa in una coda condivisa*

### **Spostamento di una coda condivisa in una coda non condivisa**

La procedura per lo spostamento di una coda condivisa in una coda non condivisa è come la procedura per lo spostamento di una coda da una struttura CF ad un'altra (consultare [“Spostamento di una coda da una struttura CFS a un'altra”](#) a pagina 496).

Figura 40 a pagina 498 fornisce un lavoro di esempio per eseguire questa operazione.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
//      DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW) QSGDISP(QMGR)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*

```

*Figura 40. Lavoro di esempio per lo spostamento di una coda condivisa in una coda non condivisa*

### **Migrazione di code non condivise in code condivise**

Esistono due fasi per la migrazione di code non condivise in code condivise:

- Migrazione del primo (o unico) gestore code nel gruppo di condivisione code
- Migrazione di altri gestori code nel gruppo di condivisione code

#### **Migrazione del primo (o unico) gestore code nel gruppo di condivisione code**

Figura 39 a pagina 498 mostra un lavoro di esempio per spostare una coda non condivisa in una coda condivisa. Effettuare questa operazione per ogni coda che deve essere migrata.

**Nota:**

1. I messaggi sulle code condivise sono soggetti a determinate restrizioni relative alla dimensione massima dei messaggi, alla persistenza dei messaggi e al tipo di indice della coda, pertanto potrebbe non essere possibile spostare alcune code non condivise in una coda condivisa.
2. È necessario utilizzare il tipo di indice corretto per le code condivise. Se si migra una coda di trasmissione in modo che sia una coda condivisa, il tipo di indice deve essere MSGID.

Se la coda è vuota o non è necessario conservare i messaggi su di essa, la migrazione della coda è più semplice. [Figura 41 a pagina 499](#) mostra un lavoro di esempio da utilizzare in queste circostanze.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqua1.SCSQANLE,DISP=SHR
// DD DSN=thlqua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(QUEUE_TO_MOVE) PUT(ENABLED) GET(ENABLED)
DELETE QL(QUEUE_TO_MOVE)
DEFINE QL(QUEUE_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
DELETE QL(TEMP_QUEUE)
/*
```

*Figura 41. Lavoro di esempio per lo spostamento di una coda non condivisa senza messaggi in una coda condivisa*

### Migrazione di altri gestori code nel gruppo di condivisione code

1. Per ogni coda che non ha lo stesso nome di una coda condivisa esistente, spostare la coda come descritto in [Figura 39 a pagina 498](#) o [Figura 41 a pagina 499](#).
2. Per le code con lo stesso nome di una coda condivisa esistente, spostare i messaggi nella coda condivisa utilizzando i comandi riportati in [Figura 42 a pagina 499](#).

```
MOVE QLOCAL(QUEUE_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(QUEUE_TO_MOVE)
DELETÉ QLOCAL(QUEUE_TO_MOVE) QSGDISP(QMGR)
```

*Figura 42. Spostamento di messaggi da una coda non condivisa a una coda condivisa esistente*

### Sospensione di una connessione a Db2

Se si desidera applicare la manutenzione o il servizio alle tabelle o ai package Db2 correlati alle code condivise senza arrestare il gestore code, è necessario disconnettere temporaneamente i gestori code nel DSG (data sharing group) da Db2.

Per far ciò:

1. Utilizzare il comando MQSC [SUSPEND QMGR FACILITY](#) ( Db2 ).
2. Eseguire i bind.
3. Riconnettersi a Db2 utilizzando il comando MQSC [RESUME QMGR FACILITY](#) ( Db2 )

Notare che esistono delle limitazioni sull'utilizzo di questi comandi.



**Attenzione:** Mentre la connessione Db2 viene sospesa, le seguenti operazioni non saranno disponibili. Pertanto, è necessario eseguire questo lavoro durante un periodo in cui l'azienda è almeno occupata.

- Accesso agli oggetti della coda condivisa per la gestione (definisci, elimina, modifica)

- Avvio dei canali condivisi
- Memorizzazione dei messaggi in Db2
- Backup o ripristino CFSTRUCT

## **Gestione degli oggetti del gruppo**

Utilizzare questo argomento per comprendere come gestire gli oggetti gruppo.

IBM MQ copia automaticamente la definizione di un oggetto gruppo nella serie di pagine zero di ciascun gestore code che lo utilizza. È possibile modificare temporaneamente la copia della definizione e IBM MQ consente di aggiornare le copie della serie di pagine dalla copia di repository. IBM MQ tenta sempre di aggiornare le copie della serie di pagine dalla copia del repository all'avvio (per gli oggetti del canale, questa operazione viene eseguita al riavvio dell'iniziatore del canale). Ciò garantisce che le copie della serie di pagine riflettano la versione sul repository, incluse le modifiche apportate quando il gestore code era inattivo.

Ci sono circostanze in cui l'aggiornamento non viene eseguito, ad esempio:

- Se una copia della coda è aperta, un aggiornamento che modificherebbe l'utilizzo della coda ha esito negativo.
- Se una copia di una coda contiene messaggi, un aggiornamento che eliminerebbe tale coda non riesce.

In queste circostanze, l'aggiornamento non viene eseguito su tale copia, ma viene eseguito sulle copie su tutti gli altri gestori code. Ricercare e correggere eventuali problemi con gli oggetti copia dopo l'aggiunta, la modifica o l'eliminazione di un oggetto gruppo e al riavvio del gestore code o dell'iniziatore del canale.

## **Gestione della CF (coupling facility)**

Utilizzare questo argomento per comprendere come aggiungere o rimuovere le strutture CF (coupling facility).

Questa sezione descrive le seguenti attività:

- [“Aggiunta di una struttura CFS \(coupling facility structure\)” a pagina 500](#)
- [“Rimozione di una struttura CFS \(coupling facility structure\)” a pagina 500](#)

### **Aggiunta di una struttura CFS (coupling facility structure)**

Per aggiungere una struttura CFS, utilizzare la seguente procedura:

1. Definire la struttura CF nel dataset della politica CFRM. Le informazioni sull'impostazione della CF (Coupling Facility) in [Impostazione della CF](#) descrivono le regole per la denominazione delle strutture CF (Coupling Facility) e come definire le strutture nel dataset della politica CFRM.
2. Se si desidera configurare la struttura per scaricare i dati del messaggio in SMDS, allocare e preformattare i dataset. Per i dettagli, consultare [creazione di un dataset di messaggi condiviso](#).
3. Definire la struttura in IBM MQ utilizzando il comando [DEFINE CFSTRUCT](#).

### **Rimozione di una struttura CFS (coupling facility structure)**

Per rimuovere una struttura CFS, utilizzare la seguente procedura:

1. Utilizzare il comando riportato di seguito per ottenere un elenco di tutte le code che utilizzano la struttura CFS che si desidera eliminare:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Eliminare tutte le code che utilizzano la struttura.
3. Eliminare la struttura CF da IBM MQ utilizzando il comando `DELETE CFSTRUCT` .
4. Se la struttura è stata configurata per scaricare i dati del messaggio in SMDS, eliminare l'SMDS.
5. Rimuovere la definizione della struttura dal dataset della politica CFRM ed eseguire il programma di utilità IXCMIAPU. (Questa è l'inverso dell'impostazione dell'attività di personalizzazione della CF (Coupling Facility), descritta in [Impostazione della CF.](#))

### **z/OS** **Ottimizzazione del monitoraggio dell'elenco CF (Coupling Facility)**

Utilizzare questo argomento per comprendere il monitoraggio dell'elenco CF (Coupling Facility)

Il monitoraggio dell'elenco CF (Coupling Facility) viene utilizzato per monitorare lo stato delle strutture dell'elenco che contengono code condivise IBM MQ . Quando un messaggio viene aggiunto a una coda condivisa e la profondità della coda passa da zero a diverso da zero, la CF notifica a tutti i gestori code nel gruppo di condivisione code. Quando viene inviata una notifica ai gestori code, è possibile eseguire una serie di azioni, inclusi i controlli dei trigger di notifica che utilizzano TRIGGER (PRIMO) o le applicazioni che stanno eseguendo un'operazione get - wait.

Per impostazione predefinita, la CF notifica tutti i gestori code nel gruppo di condivisione code contemporaneamente. In alcune configurazioni ciò può causare problemi, come:

- Distribuzione del carico di lavoro distorta, in cui una grande percentuale di messaggi viene indirizzata a un determinato gestore code nel gruppo di condivisione code, spesso il gestore code in esecuzione sulla LPAR più veloce o che è più vicino alla CF oppure
- Un numero elevato di richiami non riusciti, con conseguente perdita di tempo CPU.

z/OS V2R3 introduce un nuovo attributo CFRM (coupling facility resource manager) denominato **KEYRNOTIFYDELAY**, che può essere utilizzato con strutture di elenco contenenti code condivise (ossia, strutture dell'applicazione e non la struttura di gestione) e che può, per determinati carichi di lavoro, ridurre al minimo gli effetti delle chiamate MQGET vuote e asimmetriche del carico di lavoro o delle chiamate MQGET vuote.

**KEYRNOTIFYDELAY** può essere impostato solo su strutture in un CF, in esecuzione a CFLEVEL 22 o superiore.

Il valore deve essere composto da una a sette cifre decimali, in un intervallo compreso tra 0 e 1.000.000 microsecondi. Se impostato su un valore diverso da zero e la profondità di una transizione da zero a diverso da zero, la CF seleziona un singolo gestore code dal gruppo di condivisione code e notifica tale gestore code prima di tutti gli altri gestori code del gruppo.

Il gestore code viene selezionato in modo round - robin. Se il gestore code selezionato non elabora il messaggio all'interno dell'intervallo di tempo descritto da **KEYRNOTIFYDELAY** , verranno notificati anche tutti gli altri gestori code nel gruppo di condivisione code.

Ulteriori informazioni su **KEYRNOTIFYDELAY** sono disponibili qui: [Understanding Keyrange Monitoring Notification Delay.](#)

Notare che esistono due attributi CFRM simili denominati **LISTNOTIFYDELAY** e **SUBNOTIFYDELAY** . Nessuno di questi ha un effetto misurabile sul carico di lavoro IBM MQ .

### **z/OS** **Ripristino e riavvio su z/OS**

Utilizzare questo argomento per comprendere i meccanismi di ripristino e riavvio utilizzati da IBM MQ.

#### **z/OS** **Riavvio IBM MQ**

Una volta terminato un gestore code, sono necessarie diverse procedure di riavvio, a seconda di come è stato terminato il gestore code. Utilizzare questo argomento per comprendere le differenti procedure di riavvio che possono essere utilizzate.

Questo argomento contiene informazioni su come riavviare il gestore code nelle seguenti circostanze:

- [“Riavvio dopo un arresto normale” a pagina 502](#)
- [“Riavvio dopo una chiusura anomala” a pagina 502](#)
- [“Riavvio in caso di perdita delle serie di pagine” a pagina 502](#)
- [“Riavvio in caso di perdita dei dataset di log” a pagina 502](#)
- [Riavvio in caso di perdita delle strutture CF](#)

## Riavvio dopo un arresto normale

Se il gestore code è stato arrestato con il comando STOP QMGR, il sistema termina il suo lavoro in modo ordinato e prende un punto di controllo di terminazione prima di arrestarlo. Quando si riavvia il gestore code, esso utilizza le informazioni dal punto di controllo del sistema e dal log di recupero per determinare lo stato del sistema all'arresto.

Per riavviare il gestore code, immettere il comando START QMGR come descritto in [“Avvio e arresto di un gestore code su z/OS” a pagina 431](#).

## Riavvio dopo una chiusura anomala

IBM MQ rileva automaticamente se il riavvio segue un normale arresto o una chiusura anomala.

L'avvio del gestore code dopo che è stato terminato in modo anomalo è diverso dall'avvio dopo l'immissione del comando STOP QMGR. Se il gestore code viene terminato in modo anomalo, viene terminato senza poter terminare il lavoro o eseguire un punto di controllo di terminazione.

Per riavviare il gestore code, immettere il comando START QMGR come descritto in [“Avvio e arresto di un gestore code su z/OS” a pagina 431](#). Quando si riavvia un gestore code dopo una chiusura anomala, esso aggiorna la conoscenza del suo stato al termine utilizzando le informazioni nel log e notifica lo stato di varie attività.

Normalmente, il processo di riavvio risolve tutti gli stati incongruenti. Tuttavia, in alcuni casi, è necessario adottare misure specifiche per risolvere le incongruenze. Ciò è descritto in [“Ripristino manuale delle unità di lavoro” a pagina 516](#).

## Riavvio in caso di perdita delle serie di pagine

Se le serie di pagine sono state perse, è necessario ripristinarle dalle copie di backup prima di riavviare il gestore code. Ciò è descritto in [“Come eseguire il backup e il ripristino delle serie di pagine” a pagina 486](#).

Il riavvio del gestore code potrebbe richiedere molto tempo in queste circostanze a causa del tempo necessario per il ripristino del supporto.

## Riavvio in caso di perdita dei dataset di log

Se, dopo aver arrestato un gestore code (utilizzando il comando STOP QMGR), entrambe le copie del log vengono perse o danneggiate, è possibile riavviare il gestore code se si dispone di un set di pagine congruente (prodotto utilizzando [Metodo 1: backup completo](#)).

Seguire questa procedura:

1. Definire nuove serie di pagine che corrispondano a ciascuna serie di pagine esistente nel gestore code. Consultare [Attività 15: definire le serie di pagine](#) per informazioni sulla definizione di serie di pagine. Assicurarsi che ogni nuova serie di pagine sia più grande della corrispondente serie di pagine di origine.
2. Utilizzare la funzione FORMAT di CSQUTIL per formattare la serie di pagina di destinazione. Per ulteriori dettagli, consultare [Formattazione delle serie di pagine](#).

3. Utilizzare la funzione RESETPAGE di CSQUTIL per copiare le serie di pagine esistenti o reimpostarle e reimpostare l'RBA di log in ogni pagina. Consultare [Copia di una serie di pagine e reimpostazione del log](#) per ulteriori informazioni su questa funzione.
4. Ridefinire i dataset del log del gestore code e BSDS utilizzando CSQJU003 (consultare [Programma di utilità di inventario del log delle modifiche](#)).
5. Riavviare il gestore code utilizzando le nuove serie di pagine. Per eseguire questa operazione, effettuare una delle seguenti operazioni:
  - Modificare la procedura dell'attività avviata del gestore code per fare riferimento alle nuove serie di pagine. Per ulteriori informazioni, consultare [Attività 6: Creazione di procedure per il IBM MQ gestore code](#).
  - Utilizzare Access Method Services per eliminare i vecchi set di pagine e ridenominare i nuovi set di pagine, assegnando loro gli stessi nomi dei vecchi set di pagine.

**Attenzione:** Prima di eliminare una serie di pagine IBM MQ, assicurarsi di aver eseguito le copie di backup richieste.

Se il gestore code è un membro di un gruppo di condivisione code, le definizioni di oggetti GROUP e SHARED non sono normalmente interessate da log persi o danneggiati. Tuttavia, se i messaggi della coda condivisa sono coinvolti in un'unità di lavoro coperta dai log persi o danneggiati, l'effetto su tali messaggi non sottoposti a commit è imprevedibile.

**Nota:** Se i log sono danneggiati e il gestore code è un membro di un gruppo di condivisione code, la possibilità di recuperare i messaggi persistenti condivisi potrebbe andare persa. Emettere un comando BACKUP CFSTRUCT immediatamente su un altro gestore code attivo nel gruppo di condivisione code per tutte le strutture CF con l'attributo RECOVER (YES).

## Riavvio in caso di perdita delle strutture CF

Non è necessario riavviare se si perdono le strutture CF, perché il gestore code non viene terminato.

### **Ripristino sito alternativo su z/OS**

È possibile ripristinare un singolo gestore code o un gruppo di condivisione code oppure considerare il mirroring del disco.

Per ulteriori dettagli, consultare le seguenti sezioni:

- [Ripristino di un singolo gestore code in un sito alternativo](#)
- [Ripristino di un gruppo di condivisione code](#).
  - [Ripristino supporto struttura CF](#)
  - [Backup del gruppo di condivisione code sul sito principale](#)
  - [Ripristino di un gruppo di condivisione code sul sito alternativo](#)
- [Utilizzo del mirroring del disco](#)

## Ripristino di un singolo gestore code in un sito alternativo

Se si verifica una perdita totale di un IBM MQ computing center, è possibile eseguire il ripristino su un altro gestore code o su un gruppo di condivisione code su un sito di ripristino. (Consultare [“Ripristino di un gruppo di condivisione code sul sito alternativo”](#) a pagina 507 per la procedura alternativa di ripristino del sito per un gruppo di condivisione code.)

Per eseguire il ripristino su un altro gestore code su un sito di ripristino, è necessario eseguire regolarmente il backup delle serie di pagine e dei log. Come per tutte le operazioni di recupero dati, gli obiettivi del recupero di emergenza sono di perdere il minor numero possibile di dati, elaborazione del carico di lavoro (aggiornamenti) e tempo.

Nel sito di recupero:

- I gestori code di ripristino **devono** avere gli stessi nomi dei gestori code persi.

- Il modulo dei parametri di sistema (ad esempio, CSQZPARM) utilizzato su ogni gestore code di recupero deve contenere gli stessi parametri del corrispondente gestore code perso.

Una volta eseguita questa operazione, ristabilire tutti i gestori code come descritto nella seguente procedura. Può essere utilizzato per eseguire il ripristino di emergenza sul sito di ripristino per un singolo gestore code. Si presume che tutto ciò che è disponibile sia:

- Copie dei log di archivio e BSDS creati dalla normale esecuzione sul sito primario (i log attivi andranno persi insieme al gestore code sul sito primario).
- Le copie delle serie di pagine dal gestore code sul sito primario che hanno la stessa età o sono più vecchie delle più recenti copie di log di archivio disponibili.

È possibile utilizzare la registrazione doppia per i log attivi e di archivio, nel qual caso è necessario applicare gli aggiornamenti BSDS ad entrambe le copie:

1. Definire nuovi dataset di serie di pagine e caricarli con i dati nelle copie dei set di pagine dal sito primario.
2. Definire nuovi dataset di log attivi.
3. Definire un nuovo dataset BSDS e utilizzare Access Method Services REPRO per copiare in esso il BSDS archiviato più recente.
4. Utilizzare il programma di utilità di stampa della mappa di log CSQJU004 per stampare le informazioni da questo BSDS più recente. Quando questo BSDS è stato archiviato, il log archiviato più recente è stato troncato come log attivo e non viene visualizzato come log archiviato. Registrare STARTRBA e ENDRBA di questo log.
5. Utilizzare il programma di utilità di inventario del log delle modifiche, CSQJU003, per registrare questo ultimo dataset del log di archivio nel BSDS appena ripristinato, utilizzando STARTRBA e ENDRBA registrati al passo “4” a pagina 504.
6. Utilizzare l'opzione DELETE di CSQJU003 per rimuovere tutte le informazioni di log attive da BSDS.
7. Utilizzare l'opzione NEWLOG di CSQJU003 per aggiungere i log attivi a BSDS, non specificare STARTRBA o ENDRBA.
8. Utilizzare CSQJU003 per aggiungere un record di controllo riavvio a BSDS. Specificare CRESTART CREATE, ENDRBA=highrba, dove highrba è l'RBA elevato del log di archiviazione più recente disponibile (disponibile al passo “4” a pagina 504), più 1.  
Il BSDS ora descrive tutti i log attivi come vuoti, tutti i log archiviati disponibili e nessun punto di controllo oltre la fine dei log.
9. Riavviare il gestore code con il comando START QMGR. Durante l'inizializzazione, viene emesso un messaggio di risposta dell'operatore come il seguente:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.
REPLY Y TO CONTINUE, N TO CANCEL
```

Immettere Y per avviare il gestore code. Il gestore code viene avviato e recupera i dati fino a ENDRBA specificato nell'istruzione CRESTART.

Consultare [Utilizzo dei programmi di utilità IBM MQ](#) per informazioni sull'utilizzo di CSQJU003 e CSQJU004.

Il seguente esempio mostra istruzioni di input di esempio per CSQJU003 per i passi 6, 7 e 8:

```
* Step 6
DELETE DSNAME=MQM2.LOGCOPY1.DS01
DELETE DSNAME=MQM2.LOGCOPY1.DS02
DELETE DSNAME=MQM2.LOGCOPY1.DS03
DELETE DSNAME=MQM2.LOGCOPY1.DS04
DELETE DSNAME=MQM2.LOGCOPY2.DS01
DELETE DSNAME=MQM2.LOGCOPY2.DS02
DELETE DSNAME=MQM2.LOGCOPY2.DS03
DELETE DSNAME=MQM2.LOGCOPY2.DS04

* Step 7
```

```
NEWLOG DSNAME=MQM2.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS03,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS04,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY2.DS01,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS02,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS03,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS04,COPY2
```

```
* Step 8
CRESTART CREATE,ENDRBA=063000
```

Gli elementi da considerare per riavviare l'inziatore di canali sul sito di recupero sono simili a quelli affrontati quando si utilizza ARM per riavviare l'inziatore di canali su una diversa immagine z/OS. Per ulteriori informazioni, consultare [“Utilizzo di ARM in una rete IBM MQ”](#) a pagina 513. La strategia di recupero deve coprire anche il recupero delle librerie del prodotto IBM MQ e degli ambienti di programmazione dell'applicazione che utilizzano IBM MQ (CICS, ad esempio).

Altre funzioni del programma di utilità di inventario del log delle modifiche (CSQJU003) possono essere utilizzate anche in scenari di ripristino di emergenza. La funzione HIGHRBA consente l'aggiornamento dei valori di RBA scritti e di RBA scaricati più elevati all'interno del dataset bootstrap. La funzione CHECKPT consente l'aggiunta di nuovi record di coda di checkpoint o la cancellazione di record di coda di checkpoint esistenti in BSDS.

**Attenzione: Queste funzioni potrebbero influire sull'integrità dei dati IBM MQ.** Utilizzarli solo in scenari di ripristino di emergenza sotto la guida del personale di servizio IBM.

### Tecniche di copia rapida

Se le copie di tutte le serie di pagine e i log vengono eseguiti mentre il gestore code è bloccato, le copie saranno un set congruente che può essere utilizzato per riavviare il gestore code in un sito alternativo. In genere, abilitano un riavvio molto più rapido del gestore code, poiché è necessario eseguire un piccolo ripristino dei supporti.

Utilizzare il comando SUSPEND QMGR LOG per bloccare il gestore code. Questo comando scarica i pool di buffer nelle serie di pagine, prende un punto di controllo e arresta qualsiasi ulteriore attività di scrittura del log. Una volta sospesa l'attività di scrittura del log, il gestore code viene effettivamente bloccato fino a quando non si immette un comando RESUME QMGR LOG. Mentre il gestore code è bloccato, è possibile copiare le serie di pagine e i log.

Utilizzando gli strumenti di copia come FLASHCOPY o SNAPSHOT per copiare rapidamente le serie di pagine e i log, il tempo durante il quale il gestore code viene bloccato può essere ridotto al minimo.

All'interno di un gruppo di condivisione code, tuttavia, il comando SUSPEND QMGR LOG potrebbe non essere una buona soluzione. Per essere efficaci, le copie dei log devono contenere tutte lo stesso punto temporale per il ripristino, il che significa che il comando SUSPEND QMGR LOG deve essere emesso su tutti i gestori code all'interno del gruppo di condivisione code contemporaneamente e quindi l'intero gruppo di condivisione code verrà bloccato per un certo periodo di tempo.

### Ripristino di un gruppo di condivisione code

In caso di emergenza di un sito principale, è possibile riavviare un gruppo di condivisione code su un sito remoto utilizzando i dataset di backup dal sito principale. Per ripristinare un gruppo di condivisione code, è necessario coordinare il ripristino in tutti i gestori code nel gruppo di condivisione code e coordinarlo con altre risorse, principalmente Db2. Questa sezione descrive queste attività in modo dettagliato.

- [Ripristino supporto struttura CF](#)
- [Backup del gruppo di condivisione code sul sito principale](#)
- [Ripristino di un gruppo di condivisione code sul sito alternativo](#)

### Ripristino supporto struttura CF

Il ripristino dei supporti di una struttura CF utilizzata per conservare i messaggi persistenti su una coda condivisa, si basa su un backup dei supporti che possono essere inoltrati ripristinati dall'applicazione degli aggiornamenti registrati. Eseguire periodicamente i backup delle strutture CF utilizzando il comando MQSC BACKUP CFSTRUCT. Tutti gli aggiornamenti alle code condivise (MQGETe MQPUT) vengono scritti nel log del gestore code in cui viene eseguito l'aggiornamento. Per eseguire il recupero dei supporti di una struttura CF, è necessario applicare gli aggiornamenti registrati a tale backup dai log di tutti i gestori code che hanno utilizzato tale struttura CF. Quando si utilizza il comando MQSC RECOVER CFSTRUCT, IBM MQ unisce automaticamente i log dai gestori code rilevanti e applica gli aggiornamenti al backup più recente.

Il backup della struttura CF viene scritto nel log del gestore code che ha elaborato il comando BACKUP CFSTRUCT, quindi non ci sono ulteriori dataset da raccogliere e trasportare nel sito alternativo.

### Backup del gruppo di condivisione code sul sito principale

Sul sito principale è necessario stabilire una serie coerente di backup su base regolare, che può essere utilizzata in caso di emergenza per ricostruire il gruppo di condivisione code su un sito alternativo. Per un singolo gestore code, il ripristino può avvenire in un momento temporale arbitrario, in genere alla fine dei log disponibili sul sito remoto. Tuttavia, se i messaggi persistenti sono stati memorizzati su una coda condivisa, i log di tutti i gestori code nel gruppo di condivisione code devono essere uniti per ripristinare le code condivise, poiché qualsiasi gestore code nel gruppo di condivisione code potrebbe aver eseguito aggiornamenti ( MQPUT o MQGET ) sulla coda.

Per il ripristino di un gruppo di condivisione code, è necessario stabilire un punto temporale che rientri nell'intervallo di log dei dati di log di tutti i gestori code. Tuttavia, poiché è possibile solo **inoltrare** il supporto di ripristino dal log, questo momento temporale deve essere successivo all'immissione del comando BACKUP CFSTRUCT e dopo l'esecuzione di qualsiasi backup della serie di pagine. (In genere, il punto temporale per il ripristino potrebbe corrispondere alla fine di un giorno lavorativo o di una settimana.)

Il seguente diagramma mostra le linee temporali per due gestori code in un gruppo di condivisione code. Per ciascun gestore code, vengono eseguiti backup incompleti delle serie di pagine (consultare [Metodo 2: backup inesatto](#) ). Sul gestore code A, viene emesso un comando BACKUP CFSTRUCT. Successivamente, viene immesso un comando ARCHIVE LOG su ciascun gestore code per troncatura il log attivo e copiarlo sul supporto offline dal gestore code, che può essere trasportato al sito alternativo. La fine del log identifica l'ora in cui è stato emesso il comando ARCHIVE LOG e quindi contrassegna l'estensione dei dati di log generalmente disponibili nel sito alternativo. Il punto nel tempo per il ripristino deve essere compreso tra la fine di qualsiasi backup della serie di pagine o della struttura CF e la prima fine del log disponibile sul sito alternativo.

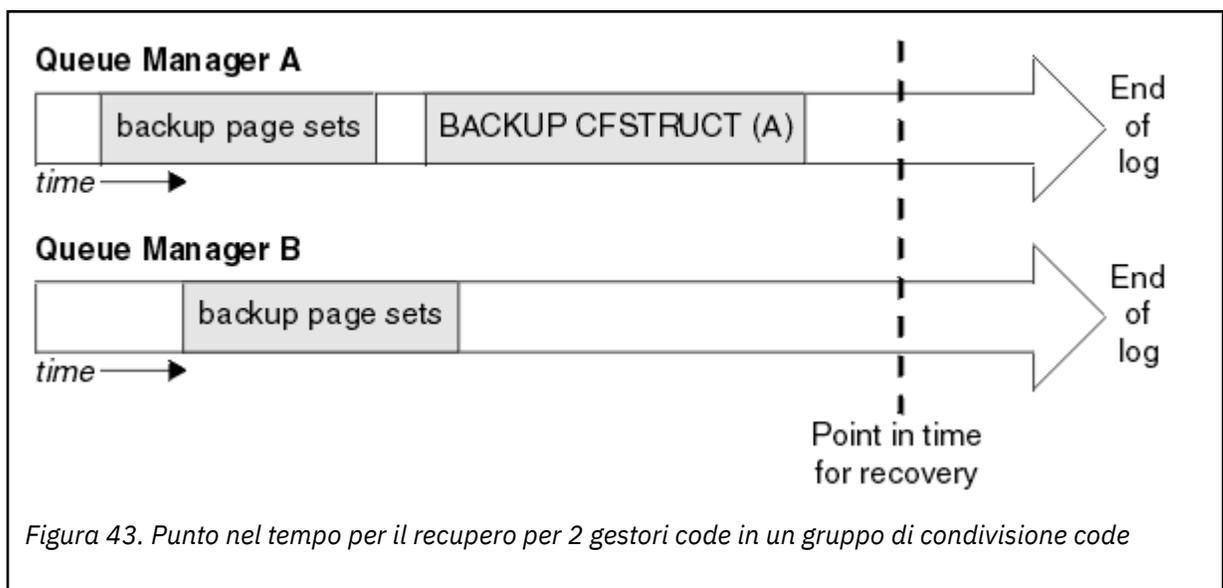


Figura 43. Punto nel tempo per il recupero per 2 gestori code in un gruppo di condivisione code

IBM MQ registra le informazioni associate ai backup della struttura CF in una tabella in Db2. A seconda dei requisiti, è possibile che si desideri coordinare il momento temporale per il ripristino di IBM MQ con quello di Db2, oppure potrebbe essere sufficiente eseguire una copia del CSQ di IBM MQ CSQ.ADMIN\_B\_STRBACKUP dopo il completamento dei comandi BACKUP CFSTRUCT.

Per preparare un ripristino:

1. Creare backup di serie di pagine per ciascun gestore code nel gruppo di condivisione code.
2. Immettere un comando BACKUP CFSTRUCT per ogni struttura CF con attributo RECOVER (YES). È possibile immettere questi comandi da un singolo gestore code o da diversi gestori code all'interno del gruppo di condivisione code per bilanciare il workload.
3. Una volta completati tutti i backup, immettere un comando ARCHIVE LOG per commutare il log attivo e creare copie dei log e dei BSDS di ciascun gestore code nel gruppo di condivisione code.
4. Trasportare i backup della serie di pagine, i file di log archiviati, i BSDS archiviati di tutti i gestori code nel gruppo di condivisione code e le informazioni di backup Db2 , offsite.

### Ripristino di un gruppo di condivisione code sul sito alternativo

Prima di poter ripristinare il gruppo di condivisione code, è necessario preparare l'ambiente:

1. Se si dispone di informazioni obsolete nella CF (Coupling Facility) dagli avvii di pratica quando è stato installato il gruppo di condivisione code, è necessario eliminare prima queste informazioni:

**Nota:** Se non si dispone di informazioni obsolete nella CF, è possibile omettere questo passo.

- a. Immettere il seguente comando z/OS per visualizzare le strutture CF per questo gruppo di condivisione code:

```
D XCF,STRUCTURE,STRNAME= qsgrname
```

- b. Per tutte le strutture che iniziano con il nome del gruppo di condivisione code, utilizzare il comando z/OS SETXCF FORCE CONNECTION per forzare la connessione fuori da tali strutture:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Eliminare tutte le strutture CF utilizzando il seguente comando per ogni struttura:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Ripristinare i sistemi Db2 e i gruppi di condivisione dati.
3. Ripristinare il CSQ CSQ.ADMIN\_B\_STRBACKUP in modo che contenga informazioni sui backup di struttura più recenti eseguiti sul sito principale.

**Nota:** È importante che la tabella STRBACKUP contenga le informazioni di backup della struttura più recenti. Le informazioni di backup della struttura meno recenti potrebbero richiedere dataset che sono stati eliminati come risultato delle informazioni fornite da un comando DISPLAY USAGE TYPE (DATASET) recente, il che significa che la struttura CF ripristinata non contiene informazioni accurate.

4. Eseguire il comando ADD QMGR del programma di utilità ... CSQ5PQSG per ogni gestore code nel gruppo di condivisione code. Questa operazione ripristinerà la voce del gruppo XCF per ogni gestore code.

Quando si esegue il programma di utilità in questo scenario, i seguenti messaggi sono normali:

```
CSQU566I Unable to get attributes for admin structure, CF not found  
or not allocated  
CSQU546E Unable to add QMGR queue_manager_name entry,  
already exists in DB2 table CSQ.ADMIN_B_QMGR  
CSQU148I CSQ5PQSG Utility completed, return code=4
```

Per ripristinare i gestori code nel gruppo di condivisione code:

1. Definire nuovi dataset di serie di pagine e caricarli con i dati nelle copie dei set di pagine dal sito primario.
2. Definire nuovi dataset di log attivi.
3. Definire un nuovo dataset BSDS e utilizzare Access Method Services REPRO per copiare in esso il BSDS archiviato più recente.
4. Utilizzare il programma di utilità di stampa della mappa di log CSQJU004 per stampare le informazioni da questo BSDS più recente. Quando questo BSDS è stato archiviato, il log archiviato più recente è stato troncato come log attivo e non viene visualizzato come log archiviato. Annotare i valori STARTRBA, STARTLRSN, ENDRBA e ENDLRSN di questo log.
5. Utilizzare il programma di utilità di inventario del log di modifica, CSQJU003, per registrare questo ultimo dataset del log di archiviazione in BSDS appena ripristinato, utilizzando i valori registrati nel Passo "4" a pagina 508.
6. Utilizzare l'opzione DELETE di CSQJU003 per rimuovere tutte le informazioni di log attive da BSDS.
7. Utilizzare l'opzione NEWLOG di CSQJU003 per aggiungere i log attivi a BSDS, non specificare STARTRBA o ENDRBA.
8. Calcolare il *recoverylrsn* per il gruppo di condivisione code. *recoverylrsn* è il valore più basso tra i ENDLRSN tra tutti i gestori code nel gruppo di condivisione code (come registrato nel Passo "4" a pagina 508), meno 1. Ad esempio, se sono presenti due gestori code nel gruppo di condivisione code e ENDLRSN per uno di essi è B713 3C72 22C5e per l'altro è B713 3D45 2123, *recoverylrsn* è B713 3C72 22C4.
9. Utilizzare CSQJU003 per aggiungere un record di controllo riavvio a BSDS. Specifica:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

dove *recoverylrsn* è il valore registrato nel passo "8" a pagina 508.

Il BSDS ora descrive tutti i log attivi come vuoti, tutti i log archiviati disponibili e nessun punto di controllo oltre la fine dei log.

È necessario aggiungere il record CRESTART al BSDS per ciascun gestore code all'interno del gruppo di condivisione code.

10. Riavviare ogni gestore code nel gruppo di condivisione code con il comando START QMGR. Durante l'inizializzazione, viene emesso un messaggio di risposta dell'operatore come il seguente:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Rispondere Y per avviare il gestore code. Il gestore code viene avviato e recupera i dati fino a ENDRBA specificato nell'istruzione CRESTART.

Per IBM WebSphere MQ 7.0.1 e versioni successive, il primo gestore code avviato può ricreare le partizioni della struttura di gestione per gli altri membri del gruppo di condivisione code e per il proprio, e non è più necessario riavviare ciascun gestore code nel gruppo di condivisione code in questa fase.

11. Quando i dati della struttura di gestione per tutti i gestori code sono stati ricreati, immettere un comando RECOVER CFSTRUCT per ogni struttura di applicazioni CF.

Se si immette il comando RECOVER CFSTRUCT per tutte le strutture su un singolo gestore code, il processo di unione dei log viene eseguito solo una volta, quindi è più rapido rispetto all'immissione del comando su un gestore code diverso per ogni struttura CF, dove ogni gestore code deve eseguire il passo di unione dei log.

Quando l'elaborazione del riavvio condizionale viene utilizzata in un gruppo di condivisione code, IBM WebSphere MQ 7.0.1 e gestori code successivi, eseguendo la rigenerazione dell'amministratore peer,

verificare che i peer BSDS contengano lo stesso CRESTART LRSN del proprio. Ciò è necessario per garantire l'integrità della struttura di gestione ricostruita. È quindi importante riavviare altri peer nel QSG, in modo che possano elaborare le proprie informazioni CRESTART, prima del successivo riavvio incondizionato di qualsiasi membro del gruppo.

## Utilizzo del mirroring del disco

Molte installazioni utilizzano ora tecnologie di mirroring del disco come IBM Metro Mirror (precedentemente PPRC) per eseguire copie sincrone di dataset su un sito alternativo. In tali situazioni, molti dei passi dettagliati diventano non necessari poiché le serie di pagine IBM MQ e i log sul sito alternativo sono effettivamente identici a quelli sul sito principale. Quando si utilizzano tali tecnologie, la procedura per riavviare un gruppo di condivisione code in un sito alternativo può essere riepilogata come segue:

- Cancellare le strutture CF IBM MQ nel sito alternativo. (Questi spesso contengono informazioni residue provenienti da qualsiasi precedente esercizio di disaster recovery).
- Ripristinare i sistemi Db2 e tutte le tabelle nel database utilizzate dal gruppo di condivisione code IBM MQ .
- Riavviare i gestori code. Prima di IBM WebSphere MQ 7.0.1, è necessario riavviare ogni gestore code definito nel gruppo di condivisione code poiché ogni gestore code recupera la propria partizione della struttura di gestione durante il riavvio del gestore code. Dopo che ogni gestore code è stato riavviato, quelli che non si trovano sulla LPAR principale possono essere nuovamente arrestati. Per IBM WebSphere MQ 7.0.1 e versioni successive, il primo gestore code avviato ricrea le partizioni della struttura di gestione per gli altri membri del proprio gruppo di condivisione code e non è più necessario riavviare ciascun gestore code nel gruppo di condivisione code.
- Dopo che la struttura di gestione è stata ricreata, ripristinare le strutture dell'applicazione.

**V 9.2.0** IBM MQ 9.1.2e versioni successive, supporta l'utilizzo della scrittura zHyper durante la scrittura nei log attivi sottoposti a mirroring mediante Metro Mirror. zHyperLa scrittura può aiutare a ridurre l'impatto sulle prestazioni dell'utilizzo di Metro Mirror; per ulteriori informazioni, consultare [Utilizzo di Metro Mirror con IBM MQ](#) .

## **z/OS** Reinizializzazione di un gestore code

Se il gestore code è stato terminato in modo anomalo, potrebbe non essere possibile riavviarlo. Ciò potrebbe essere dovuto alla perdita, al troncamento o al danneggiamento delle serie di pagine o dei log. Se ciò si è verificato, potrebbe essere necessario reinizializzare il gestore code (eseguire un avvio a freddo).

## Attenzione

### **Eseguire un avvio a freddo solo se non è possibile riavviare il gestore code in un altro modo.**

L'esecuzione di un avvio a freddo consente di ripristinare il gestore code e le definizioni degli oggetti; **non** sarà possibile ripristinare i dati del messaggio. Verificare che nessuno degli altri scenari di riavvio descritti in questo argomento funzioni prima di eseguire questa operazione.

Una volta riavviati, tutti gli oggetti IBM MQ sono definiti e disponibili per l'uso, ma non sono presenti dati di messaggio.

**Nota:** Non reavviare un gestore code quando fa parte di un cluster. È necessario prima rimuovere il gestore code dal cluster (utilizzando i comandi RESET CLUSTER sugli altri gestori code nel cluster), quindi reinizializzare il gestore code e infine reintrodurlo nel cluster come nuovo gestore code.

Questo perché durante la reinizializzazione, l'identificativo del gestore code (QMID) viene modificato, quindi qualsiasi oggetto cluster con il vecchio identificativo del gestore code deve essere rimosso dal cluster.

Per ulteriori informazioni si vedano le seguenti sezioni:

- [Reinizializzazione di un gestore code che non si trova in un gruppo di condivisione code](#)

- Reinizializzazione dei gestori code in un gruppo di condivisione code

## **Reinizializzazione di un gestore code che non si trova in un gruppo di condivisione code**

Per reinizializzare un gestore code, seguire questa procedura:

1. Preparare le istruzioni di definizione dell'oggetto da utilizzare quando si riavvia il gestore code. Per effettuare questa operazione:
  - Se la serie di pagine zero è disponibile, utilizzare la funzione CSQUTIL SDEFS (vedere Produzione di un elenco di IBM MQ comandi di definizione ). È necessario richiamare definizioni per tutti i tipi di oggetto (oggetti delle informazioni di autenticazione, strutture CF, canali, elenchi nomi, processi, code e classi di memoria).
  - Se la serie di pagine zero non è disponibile, utilizzare le definizioni dell'ultima volta in cui è stato eseguito il backup delle proprie definizioni di oggetto.
2. Ridefinire i dataset del gestore code (non eseguire questa operazione fino a quando non è stato completato il passo “1” a pagina 510 ).

Per ulteriori informazioni, consultare creazione dei dataset bootstrap e log e definizione delle serie di pagine .

3. Riavviare il gestore code utilizzando i dataset di log appena definiti e inizializzati, BSDS e set di pagine. Utilizzare le istruzioni di input di definizione dell'oggetto create nel passo “1” a pagina 510 come input nel dataset di input di inizializzazione CSQINP2 .

## **Reinizializzazione dei gestori code in un gruppo di condivisione code**

In un gruppo di condivisione code, la reinizializzazione di un gestore code è più complessa. Potrebbe essere necessario reinizializzare uno o più gestori code a causa di problemi di serie di pagine o di log, ma potrebbero verificarsi anche problemi con Db2 o con la CF (coupling facility) da gestire. A causa di questo, ci sono una serie di alternative:

### **Avvio a sistema spento**

La reinizializzazione dell'intero gruppo di condivisione code implica la forzatura di tutte le strutture CF (Coupling Facility), la cancellazione di tutte le definizioni di oggetto per il gruppo di condivisione code da Db2, l'eliminazione o la ridefinizione dei log e dei BSDS e la formattazione delle serie di pagine per tutti i gestori code nel gruppo di condivisione code.

### **Definizioni condivise conservate**

Eliminare o ridefinire i log e BSDS, formattare le serie di pagine per tutti i gestori code nel gruppo di condivisione code e forzare tutte le strutture CF (Coupling Facility). Al riavvio, tutti i messaggi saranno stati eliminati. I gestori code ricreano gli oggetti COPY che corrispondono agli oggetti GROUP ancora esistenti nel database Db2 . Tutte le code condivise esistono ancora e possono essere utilizzate.

### **Gestore code singolo reinizializzato**

Eliminare o ridefinire i log e BSDS e formattare le serie di pagine per il singolo gestore code (questo elimina tutti i relativi oggetti e messaggi privati). Al riavvio, il gestore code ricrea gli oggetti COPY che corrispondono agli oggetti GROUP ancora esistenti nel database Db2 . Le code condivise esistono ancora, così come i messaggi su di esse, e possono essere utilizzate.

### **Recupero point - in - time di un gruppo di condivisione code**

Questo è lo scenario alternativo di ripristino di emergenza del sito.

Gli oggetti condivisi vengono ripristinati al punto temporale raggiunto dal ripristino Db2 (descritto in Un sistema Db2 ha esito negativo ). Ogni gestore code può essere ripristinato ad un punto nel tempo ottenibile dalle copie di backup disponibili sul sito alternativo.

I messaggi persistenti possono essere utilizzati nei gruppi di condivisione code e possono essere recuperati utilizzando il comando MQSC RECOVER CFSTRUCT. Notare che questo comando viene

ripristinato al momento dell'errore. Tuttavia, non esiste alcun ripristino dei messaggi della coda condivisa non persistenti; essi vengono persi a meno che non siano state eseguite copie di backup in modo indipendente utilizzando la funzione COPY del programma di utilità CSQUTIL.

Non è necessario tentare di ripristinare ogni gestore code allo stesso momento poiché non vi sono interdipendenze tra gli oggetti locali su gestori code differenti (che sono effettivamente ripristinati) e la risincronizzazione del gestore code con Db2 al riavvio crea o elimina gli oggetti COPY come necessario su un gestore code in base al gestore code.

## Utilizzo di z/OS ARM (Automatic Restart Manager)

Utilizzare questo argomento per comprendere come utilizzare ARM per riavviare automaticamente i gestori code.

Questa sezione contiene informazioni sui seguenti argomenti:

- [“Cos' è l'ARM?” a pagina 511](#)
- [“politiche ARM” a pagina 512](#)
- [“Utilizzo di ARM in una rete IBM MQ” a pagina 513](#)

### Cos' è l'ARM?

z/OS ARM (Automatic Restart Manager) è una z/OS funzione di recupero che può migliorare la disponibilità dei gestori code. Quando un lavoro o un'attività ha esito negativo o il sistema su cui è in esecuzione ha esito negativo, ARM può riavviare il lavoro o l'attività senza l'intervento dell'operatore.

Se un gestore code o un iniziatore di canali ha avuto esito negativo, ARM lo riavvia sulla stessa immagine z/OS. Se z/OS, e quindi un intero gruppo di sottosistemi e applicazioni correlati hanno avuto esito negativo, ARM può riavviare automaticamente tutti i sistemi in errore, in un ordine predefinito, su un'altra immagine z/OS all'interno del sysplex. Questo viene definito *riavvio tra sistemi*.

Riavviare l'iniziatore di canali da ARM solo in circostanze eccezionali. Se il gestore code viene riavviato da ARM, riavviare l'iniziatore di canali dal data set di inizializzazione CSQINP2 (consultare [“Utilizzo di ARM in una rete IBM MQ” a pagina 513](#)).

È possibile utilizzare ARM per riavviare un gestore code su un'immagine z/OS differente nel sysplex in caso di errore z/OS. Le implicazioni di rete del riavvio di IBM MQ ARM su un'immagine z/OS differente sono descritte in [“Utilizzo di ARM in una rete IBM MQ” a pagina 513](#).

Per abilitare il riavvio automatico:

- Impostare un set di dati di coppia di ARM.
- Definire le azioni di riavvio automatico che z/OS deve eseguire in una *politica di ARM*.
- Avviare la politica di ARM.

Inoltre, IBM MQ deve registrarsi con ARM all'avvio (ciò avviene automaticamente).

**Nota:** Se si desidera riavviare automaticamente i gestori code in immagini z/OS differenti, è necessario definire ogni gestore code come sottosistema in ogni immagine z/OS su cui tale gestore code potrebbe essere riavviato, con un nome di sottosistema univoco a livello di sysplex di quattro caratteri.

### Coppia di dataset ARM

Accertarsi di definire i dataset di coppia richiesti per ARM e che siano in linea e attivi prima di avviare qualsiasi gestore code per cui si desidera il supporto ARM. La registrazione ARM automatica di IBM MQ non riesce se i dataset di coppia non sono disponibili all'avvio del gestore code. In questa situazione, IBM MQ presume che l'assenza del dataset di coppia significhi che non si desidera il supporto ARM e l'inizializzazione continua.

Consultare [z/OS MVS Setting up a Sysplex](#) per informazioni sui dataset di coppia ARM.

## z/OS **politiche ARM**

Le politiche di Automatic Restart Manager sono regole definite dall'utente che controllano le funzioni ARM che possono controllare i riavvii di un gestore code.

Le funzioni ARM sono controllate da una *politica ARM* definita dall'utente. Ogni immagine z/OS in esecuzione su un'istanza del gestore code che deve essere riavviata da ARM deve essere connessa a un dataset di coppia ARM con una politica ARM attiva.

IBM fornisce una politica ARM predefinita. È possibile definire nuove politiche o sovrascrivere le impostazioni predefinite della politica utilizzando il *programma di utilità dati di gestione* (IXCMIAPU) fornito con z/OS. *z/OS MVS Setting up a Sysplex* descrive questo programma di utilità e include dettagli completi su come definire una politica ARM.

Figura 44 a pagina 512 mostra un esempio di politica ARM. Questa politica di esempio riavvia qualsiasi gestore code all'interno di un sysplex, se il gestore code ha avuto esito negativo o se l'intero sistema ha avuto esito negativo.

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
RESTART_GROUP(DEFAULT)
ELEMENT(*)
RESTART_ATTEMPTS(0) /* Jobs not to be restarted by ARM */
RESTART_GROUP(GROUP1)
ELEMENT(SYSQMGRM*) /* These jobs to be restarted by ARM */
/*
```

Figura 44. Politica ARM di esempio

Per ulteriori informazioni, consultare:

- [Definizione di una politica ARM](#)
- [Attivazione di una politica ARM](#)
- [Registrazione con ARM](#)

### Definizione di una politica ARM

Impostare la politica ARM nel modo seguente:

- Definire RESTART\_GROUPS per ogni istanza del gestore code che contiene anche qualsiasi sottosistema CICS o IMS che si connette a tale istanza del gestore code. Se si utilizza una convenzione di denominazione del sottosistema, è possibile utilizzare il carattere '?' e i caratteri jolly '\*' nei nomi elemento per definire RESTART\_GROUPS con minimo sforzo di definizione.
- Specificare TERMTYPE (ELEMTERM) per gli iniziatori di canali per indicare che verranno riavviati solo se l'inziatore di canali ha avuto esito negativo e l'immagine z/OS non ha avuto esito negativo.
- Specificare TERMTYPE (ALLTERM) per i gestori code per indicare che verranno riavviati se il gestore code ha avuto esito negativo o se l'immagine z/OS ha avuto esito negativo.
- Specificare RESTART\_METHOD (BOTH, PERSIST) per i gestori code e gli iniziatori di canali. Ciò indica ad ARM di riavviare utilizzando il JCL salvato (dopo la risoluzione dei simboli di sistema) durante l'ultimo avvio. Indica ad ARM di eseguire questa operazione indipendentemente dal fatto che il singolo elemento non sia riuscito o che l'immagine z/OS non sia riuscita.
- Accettare i valori predefiniti per tutte le altre opzioni della politica ARM.

### Attivazione di una politica ARM

Per avviare la politica di gestione del riavvio automatico, immettere il seguente comando z/OS :

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

Quando la politica viene avviata, tutti i sistemi connessi al dataset di coppia ARM utilizzano la stessa politica attiva.

Utilizzare il comando SETXCF STOP per disabilitare i riavvii automatici.

## Registrazione con ARM

IBM MQ si registra automaticamente come *elemento ARM* durante l'avvio del gestore code (in base alla disponibilità ARM). Annulla la registrazione durante la fase di chiusura, a meno che non sia richiesto di non farlo.

All'avvio, il gestore code determina se ARM è disponibile. In caso affermativo, IBM MQ esegue la registrazione utilizzando il nome SYSMQMGR *ssid*, dove *ssid* è il nome del gestore code di quattro caratteri e SYSMQMGR è il tipo di elemento.

I comandi STOP QMGR MODE (QUIESCE) e STOP QMGR MODE (FORCE) annullano la registrazione del gestore code da ARM (se è stato registrato con ARM all'avvio). Ciò impedisce ad ARM di riavviare questo gestore code. Il comando STOP QMGR MODE (RESTART) non annulla la registrazione del gestore code da ARM, quindi è idoneo per il riavvio automatico immediato.

Ogni spazio di indirizzo dell'iniziatore di canali determina se ARM è disponibile e, in tal caso, si registra con il nome dell'elemento SYSMQCH *ssid*, dove *ssid* è il nome del gestore code e SYSMQCH è il tipo di elemento.

L'iniziatore di canali viene sempre annullato dalla registrazione ARM quando si arresta normalmente e rimane registrato solo se termina in modo anomalo. La registrazione dell'iniziatore di canali viene sempre annullata se il gestore code ha esito negativo.

### **Utilizzo di ARM in una rete IBM MQ**

È possibile configurare il gestore code in modo che gli iniziatori di canali e i listener associati vengano avviati automaticamente al riavvio del gestore code.

Per garantire il riavvio completamente automatico del gestore code sulla stessa z/OS immagine per i protocolli di comunicazione LU 6.2 e TCP/IP:

- Avviare automaticamente i listener aggiungendo il comando START LISTENER appropriato al dataset CSQINPX.
- Avviare automaticamente l'iniziatore di canali aggiungendo il comando START CHINIT appropriato al dataset CSQINP2 .

Per riavviare un gestore code con TCP/IP o LU6.2, consultare

- [“Riavvio su un'immagine z/OS differente con TCP/IP” a pagina 513](#)
- [“Riavvio su un'altra immagine z/OS con LU 6.2” a pagina 515](#)

Consultare [Attività 13: personalizzare i dataset di input di inizializzazione](#) per informazioni sui dataset CSQINP2 e CSQINPX.

## Riavvio su un'immagine z/OS differente con TCP/IP

Se si utilizza TCP/IP come protocollo di comunicazione e si utilizzano gli indirizzi IP virtuali, è possibile configurarli per il ripristino su altre immagini z/OS , consentendo ai canali che si collegano a tale gestore code di riconnettersi senza alcuna modifica. In caso contrario, è possibile riassegnare un indirizzo TCP/IP dopo lo spostamento di un gestore code in un'immagine z/OS differente solo se si utilizzano cluster o se si

sta effettuando la connessione a un gruppo di condivisione code utilizzando un nome gruppo logico DNS (Domain Name System) dinamico WLM.

- Quando si utilizzano i cluster
- Quando ci si connette a un gruppo di condivisione code

### **Quando si utilizza il clustering**

z/OS ARM risponde a un errore di sistema riavviando il gestore code su un'immagine z/OS diversa nello stesso sysplex; questo sistema ha un indirizzo TCP/IP diverso dall'immagine z/OS originale. Quanto segue spiega come utilizzare i cluster IBM MQ per riassegnare l'indirizzo TCP/IP di un gestore code dopo che è stato spostato dal riavvio ARM a una diversa immagine z/OS .

Quando un gestore code del client rileva l'errore del gestore code (come un errore del canale), risponde riassegnando i messaggi adatti sulla coda di trasmissione del cluster a un gestore code del server differente che ospita un'istanza differente della coda cluster di destinazione. Tuttavia, non è in grado di riassegnare i messaggi che sono collegati al server originale da vincoli di affinità o i messaggi che sono in dubbio perché il gestore code del server ha riportato un errore durante l'elaborazione di fine batch. Per elaborare questi messaggi, effettuare le seguenti operazioni:

1. Assegnare un nome canale ricevente del cluster e una porta TCP/IP differenti a ogni gestore code z/OS . Ogni gestore code ha bisogno di una porta diversa in modo che due sistemi possano condividere un singolo stack TCP/IP su un'immagine z/OS . Uno di questi è il gestore code in origine in esecuzione su tale immagine z/OS e l'altro è il gestore code che ARM riavvierà su tale immagine z/OS in seguito a un malfunzionamento del sistema. Configurare ogni porta su ogni immagine z/OS , in modo che ARM possa riavviare qualsiasi gestore code su qualsiasi immagine z/OS .
2. Creare un file di input del comando dell'iniziatore di canale differente (CSQINPX) per ogni combinazione di gestore code e immagine z/OS , a cui fare riferimento durante l'avvio dell'iniziatore di canale.

Ogni file CSQINPX deve includere un comando START LISTENER PORT (port) specifico per tale gestore code e un comando ALTER CHANNEL per un canale ricevente del cluster specifico per tale gestore code e per la combinazione di immagine z/OS . Il comando ALTER CHANNEL deve impostare il nome della connessione sul nome TCP/IP dell'immagine z/OS su cui viene riavviato. Deve includere il numero di porta specifico per il gestore code riavviato come parte del nome connessione.

Il JCL di avvio di ciascun gestore code può avere un nome di dataset fisso per questo file CSQINPX e ogni immagine z/OS deve avere una versione differente di ciascun file CSQINPX su un volume DASD non condiviso.

Se si verifica un riavvio di ARM, IBM MQ annuncia la definizione di canale modificata nel repository del cluster, che a sua volta la pubblica in tutti i gestori code client che hanno espresso un interesse nel gestore code del server.

Il gestore code del client considera l'errore del gestore code del server come un errore del canale e tenta di riavviarlo. Quando il gestore code client apprende il nuovo nome - connessione server, il riavvio del canale riconnette il gestore code client al gestore code del server riavviato. Il gestore code client può quindi risincronizzare i propri messaggi, risolvere eventuali messaggi in dubbio sulla coda di trasmissione del gestore code client e la normale elaborazione può continuare.

### **Quando ci si connette a un gruppo di condivisione code**

Quando ci si connette a un gruppo di condivisione code tramite un nome gruppo logico DNS (Domain Name System) dinamico TCP/IP, il nome connessione nella definizione di canale specifica il nome gruppo logico del gruppo di condivisione code, non il nome host o l'indirizzo IP di una macchina fisica. Quando questo canale viene avviato, si connette al DNS dinamico e viene quindi connesso a un gestore code nel gruppo di condivisione code. Questo processo è descritto in Impostazione della comunicazione per IBM MQ for z/OS utilizzando i gruppi di condivisione code.

Nel caso improbabile di un errore di immagine, si verifica una delle seguenti condizioni:

- I gestori code sull'immagine in errore deregistrano dal DNS dinamico in esecuzione sul sysplex. Il canale risponde all'errore di connessione immettendo lo stato NUOVO tentativo e quindi si connette al DNS dinamico in esecuzione sul sysplex. Il DNS dinamico assegna la richiesta in entrata a uno dei restanti membri del gruppo di condivisione code ancora in esecuzione sulle immagini rimanenti.
- Se nessun altro gestore code nel gruppo di condivisione code è attivo e ARM riavvia il gestore code e l'inziatore di canali su un'immagine differente, il listener del gruppo esegue la registrazione con il DNS dinamico da questa nuova immagine. Ciò significa che il nome del gruppo logico (dal campo nome connessione del canale) si connette al DNS dinamico e viene quindi connesso allo stesso gestore code, ora in esecuzione su un'immagine diversa. Non è stata richiesta alcuna modifica alla definizione del canale.

Affinché questo tipo di recupero si verifichi, è necessario notare i punti seguenti:

- Su z/OS, il DNS dinamico viene eseguito su una delle immagini z/OS nel sysplex. Se questa immagine ha esito negativo, il DNS dinamico deve essere configurato in modo che vi sia un server dei nomi secondario attivo nel sysplex, che funga da alternativa al server dei nomi primario. Le informazioni sui server DNS dinamici primari e secondari sono disponibili nel manuale [OS/390 SecureWay CS IP Configuration](#).
- Il listener del gruppo TCP/IP potrebbe essere stato avviato su un particolare indirizzo IP che potrebbe non essere disponibile su questa immagine z/OS. In tal caso, potrebbe essere necessario avviare il listener su un indirizzo IP differente sulla nuova immagine. Se si utilizzano indirizzi IP virtuali, è possibile configurarli per il ripristino su altre immagini z/OS in modo che non sia richiesta alcuna modifica al comando START LISTENER.

## Riavvio su un'altra immagine z/OS con LU 6.2

Se si utilizzano solo i protocolli di comunicazione LU 6.2, completare la seguente procedura per abilitare la riconnessione di rete dopo il riavvio automatico di un gestore code su un'immagine z/OS differente all'interno del sysplex:

- Definire ciascun gestore code nel sysplex con un nome di sottosistema univoco.
- Definire ciascun iniziatore di canali all'interno del sysplex con un LUNAME univoco. Viene specificato sia negli attributi del gestore code che nel comando START LISTENER.

**Nota:** Il LUNAME denomina una voce nella tabella laterale APPC, che a sua volta la associa al LUNAME effettivo.

- Configurare una tabella laterale APPC condivisa, a cui fa riferimento ogni immagine z/OS all'interno del sysplex. Deve contenere una voce per ogni LUNAME dell'inziatore di canali. Consultare [z/OS MVS Planning: APPC/MVS Management](#) per informazioni su questo argomento.
- Impostare un membro APPCPM xx di SYS1.PARMLIB per ogni iniziatore di canali all'interno del sysplex per contenere un LUADD per attivare la voce della tabella laterale APPC per tale iniziatore di canali. Questi membri devono essere condivisi da ogni immagine z/OS. Il SYS1 SYS1.PARMLIB viene attivato da un comando z/OS SET APPC= xx, che viene emesso automaticamente durante il riavvio ARM del gestore code (e del suo iniziatore di canali) su un'immagine z/OS differente, come descritto nel seguente testo.
- Utilizzare l'attributo del gestore code LU62ARM per specificare il suffisso xx di SYS1.PARMLIB per ciascun iniziatore di canali. Ciò fa in modo che l'inziatore del canale immetta il comando z/OS SET APPC= xx richiesto per attivare il relativo LUNAME.

Definire la politica ARM in modo che riavvii l'inziatore di canali solo se ha esito negativo mentre la relativa immagine z/OS rimane in attività; l>ID utente associato allo spazio di indirizzo XCFAS deve essere autorizzato a immettere il comando IBM MQ START CHINIT. Non riavviare automaticamente l'inziatore di canali se anche la sua immagine z/OS ha esito negativo, utilizzare invece i comandi nei dataset CSQINP2 e CSQINPX per avviare l'inziatore di canali e i listener.

## **Ripristino manuale delle unità di lavoro**

È possibile ripristinare manualmente le unità di lavoro CICS, IMS, RRS o altri gestori code in un gruppo di condivisione code. È possibile utilizzare i comandi del gestore code per visualizzare lo stato delle unità di lavoro associate a ciascuna connessione al gestore code.

Questo argomento contiene informazioni sui seguenti argomenti:

- [“Visualizzazione di connessioni e thread” a pagina 516](#)
- [“Ripristino manuale delle unità di ripristino CICS” a pagina 516](#)
- [“Ripristino manuale delle unità di ripristino IMS” a pagina 520](#)
- [“Ripristino manuale delle unità di ripristino RRS” a pagina 522](#)
- [“Ripristino delle unità di ripristino su un altro gestore code nel gruppo di condivisione code” a pagina 522](#)

### **Visualizzazione di connessioni e thread**

È possibile utilizzare il comando `DISPLAY CONN` per ottenere informazioni sulle connessioni ai gestori code e le relative unità di lavoro associate. È possibile visualizzare le unità di lavoro attive per visualizzare ciò che sta attualmente accadendo o per visualizzare ciò che deve essere terminato per consentire l'arresto del gestore code ed è possibile visualizzare le unità di lavoro non risolte per facilitare il ripristino.

#### **Unità di lavoro attive**

Per visualizzare solo le unità di lavoro attive, utilizzare

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

#### **Unità di lavoro non risolte**

Un'unità di lavoro non risolta, nota anche come "thread in dubbio", si trova nel secondo passaggio dell'operazione di commit a due fasi. Le risorse sono conservate in IBM MQ per suo conto. Per visualizzare le unità di lavoro non risolte, utilizzare

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

L'intervento esterno è necessario per risolvere lo stato delle unità di lavoro non risolte. Ciò potrebbe comportare solo l'avvio del coordinatore di ripristino ( CICS, IMS o RRS) o potrebbe coinvolgere di più, come descritto nelle seguenti sezioni.

## **Ripristino manuale delle unità di ripristino CICS**

Utilizzare questo argomento per comprendere cosa accade quando l'adattatore CICS viene riavviato e quindi spiega come gestire le unità di ripristino non risolte che si verificano.

### **Cosa accade quando l'adattatore CICS viene riavviato**

Ogni volta che una connessione viene interrotta, l'adattatore deve passare attraverso una *fase di riavvio* durante il processo di *riconnessione*. La fase di riavvio risincronizza le risorse. La risincronizzazione tra CICS e IBM MQ consente di identificare e risolvere le unità di lavoro in dubbio.

La risincronizzazione può essere causata da:

- Una richiesta esplicita dal componente di accodamento distribuito
- Una richiesta implicita quando viene effettuata una connessione a IBM MQ

Se la risincronizzazione è causata dalla connessione a IBM MQ, la sequenza di eventi è:

1. Il processo di connessione richiama un elenco di ID UOW (units of work) in dubbio da IBM MQ.
2. Gli ID UOW vengono visualizzati sulla console nei messaggi CSQC313I .
3. Gli ID UOW vengono passati a CICS.
4. CICS avvia un'attività di risincronizzazione (CRSY) per ciascun ID UOW in dubbio.
5. Il risultato dell'attività per ogni UOW in dubbio viene visualizzato sulla console.

È necessario verificare i messaggi visualizzati durante il processo di connessione:

#### **CSQC313I**

Mostra che una UOW è in dubbio.

#### **CSQC400I**

Identifica la UOW ed è seguita da uno dei seguenti messaggi:

- CSQC402I o CSQC403I mostra che la UOW è stata risolta correttamente (con commit o backout).
- CSQC404E, CSQC405E, CSQC406E o CSQC407E indica che la UOW non è stata risolta.

#### **CSQC409I**

Mostra che tutte le UOW sono state risolte correttamente.

#### **CSQC408I**

Mostra che non tutte le UOW sono state risolte correttamente.

#### **CSQC314I**

Avverte che gli ID UOW evidenziati con un \* non vengono risolti automaticamente. Queste UOW devono essere risolte esplicitamente dal componente di accodamento distribuito quando viene riavviato.

La [Figura 45 a pagina 517](#) mostra una serie di esempio di messaggi di riavvio visualizzati sulla console z/OS .

```
CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCQCON CONNECT received from TERMID=PB62 TRANID=CKCN
+CSQC303I VICIC1 CSQCCON CSQCSERV loaded. Entry point is 850E8918
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F0E2178D25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F055B2AC25 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6EFFD60D425 is in doubt
+CSQC313I VICIC1 CSQCCON UOWID=VICIC1.A6E5A6F07AB56D22 is in doubt
+CSQC307I VICIC1 CSQCCON Successful connection to subsystem VC2
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAD18) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAA10) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BA708) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAE88) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAB80) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA878) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA570) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA268) connect
successful
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6EFFD60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully
```

*Figura 45. Messaggi di riavvio di esempio*

Il numero totale di messaggi CSQC313I deve essere uguale al numero totale di messaggi CSQC402I più CSQC403I . Se i totali non sono uguali, ci sono UOW che il processo di connessione non può risolvere. Le UOW che non possono essere risolte sono causate da problemi con CICS (ad esempio, un avvio a freddo) o con IBM MQ, oppure dalla distribuzione dell'accodamento. Quando questi problemi sono stati risolti, è possibile avviare un'altra risincronizzazione disconnettendosi e ricollegandosi.

In alternativa, è possibile risolvere personalmente ogni UOW in sospeso utilizzando il comando RESOLVE INDOUBT e l'ID UOW visualizzato nel messaggio CSQC400I. È quindi necessario avviare una disconnessione e una connessione per ripulire l' *unità di descrittori di ripristino* in CICS. È necessario conoscere il risultato corretto della UOW per risolvere le UOW manualmente.

Tutti i messaggi associati alle UOW non risolte sono bloccati da IBM MQ e nessuna attività batch, TSO o CICS può accedervi.

Se CICS ha esito negativo ed è necessario un riavvio di emergenza, *non* modificare l'APPLID GENERICO del sistema CICS . Se lo fai e poi ti riconnetti a IBM MQ, l'integrità dei dati con IBM MQ non può essere garantita. Ciò è dovuto al fatto che IBM MQ considera la nuova istanza di CICS come una CICS differente (perché l'APPLID è diverso). La risoluzione in dubbio si basa quindi sul log CICS errato.

## Come risolvere manualmente le unità di ripristino CICS

Se l'adattatore termina in modo anomalo, CICS e IBM MQ creano un elenco in dubbio in modo dinamico o durante il riavvio, a seconda del sottosistema che ha causato l'interruzione.

**Nota:** Se si utilizza il programma di esempio DFH\$INDB per visualizzare le unità di lavoro, è possibile che non vengano sempre visualizzate correttamente le UOW IBM MQ .

Quando CICS si connette a IBM MQ, potrebbero essere presenti una o più unità di ripristino che non sono state risolte.

Uno dei seguenti messaggi viene inviato alla console:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

Per dettagli sul significato di questi messaggi, consultare i messaggi dell'adattatore [CICS](#) e i messaggi [Bridge](#) .

CICS conserva i dettagli delle unità di ripristino che non sono state risolte durante l'avvio della connessione. Una voce viene eliminata quando non appare più nell'elenco presentato da IBM MQ.

Le unità di ripristino che CICS non può risolvere devono essere risolte manualmente utilizzando i comandi IBM MQ . Questa procedura manuale viene utilizzata raramente all'interno di un'installazione, poiché è richiesta solo quando errori operativi o problemi software hanno impedito la risoluzione automatica. *È necessario esaminare eventuali incongruenze rilevate durante la risoluzione in dubbio.*

Per risolvere le unità di ripristino:

1. Ottenere un elenco delle unità di ripristino da IBM MQ utilizzando il seguente comando:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

Viene ricevuto il seguente messaggio:

```

CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN (BC85772CBE3E0001)
EXTCONN (C3E2D8C3C7D9F0F940404040404040)
TYPE (CONN)
CONNOPTS (
MQCNO_STANDARD_BINDING
)
UOWLOGDA (2005-02-04)
UOWLOGTI (10.17.44)
UOWSTDA (2005-02-04)
UOWSTTI (10.17.44)
UOWSTATE (UNRESOLVED)
NID (IYRCSQ1 .BC8571519B60222D)
EXTURID (BC8571519B60222D)
QMURID (0000002BDA50)
URTYPE (CICS)
USERID (MQTEST)
APPLTAG (IYRCSQ1)
ASID (0000)
APPLTYPE (CICS)
TRANSID (GP02)
TASKNO (0000096)
END CONN DETAILS

```

Per le connessioni CICS, l'NID è composto dall'applid CICS e da un numero univoco fornito da CICS al momento della scrittura delle voci del log del punto di sincronizzazione. Questo numero univoco viene memorizzato in record scritti sia nel log di sistema CICS che nel log IBM MQ al momento dell'elaborazione del punto di sincronizzazione. Questo valore viene indicato in CICS come *token di ripristino*.

2. Eseguire la scansione del log CICS per le voci relative a una particolare unità di ripristino.

Ricerca un record PREPARE per l'installazione relativa all'attività in cui il campo del token di recupero (JCSRMTKN) è uguale al valore ottenuto dall'ID di rete. L'ID di rete viene fornito da IBM MQ nell'emissione del comando DISPLAY CONN.

Il record PREPARE nel file di log CICS per le unità di ripristino fornisce il numero attività CICS. Tutte le altre voci nel log per questa attività CICS possono essere ubicate utilizzando questo numero.

È possibile utilizzare il programma di utilità di stampa giornale CICS DFHJUP durante la scansione del log. Per dettagli sull'utilizzo di questo programma, consultare il manuale *CICS Operations and Utilities Guide*.

3. Eseguire la scansione del log IBM MQ per i record con l'NID correlato a una particolare unità di ripristino. Utilizzare quindi l'URID da questo record per ottenere il resto dei record di log per questa unità di ripristino.

Quando si esegue la scansione del log IBM MQ, si noti che il IBM MQ messaggio di avvio CSQJ001I fornisce l'RBA di avvio per questa sessione.

Il programma di stampa dei record di registrazione (CSQ1LOGP) può essere utilizzato per tale scopo.

4. Se è necessario, eseguire una risoluzione in dubbio in IBM MQ.

IBM MQ può essere indirizzato per eseguire l'azione di ripristino per un'unità di ripristino utilizzando un comando IBM MQ [RESOLVE INDOUBT](#).

Per ripristinare tutti i thread associati ad un *nome - connessione* specifico, utilizzare l'opzione NID (\*).

Il comando genera uno dei seguenti messaggi che indica se il thread è stato sottoposto a commit o a backout:

```

CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED

```

Quando si esegue una risoluzione in dubbio, CICS e l'adattatore non sono a conoscenza dei comandi per IBM MQ per eseguire il commit o il backout delle unità di recupero, poiché solo le risorse IBM MQ sono interessate. Tuttavia, CICS conserva i dettagli relativi ai thread in dubbio che non possono essere risolti da IBM MQ. Queste informazioni vengono eliminate quando l'elenco presentato è vuoto o quando l'elenco non include un'unità di recupero di cui CICS dispone di dettagli.

## **Ripristino manuale delle unità di ripristino IMS**

Utilizzare questo argomento per comprendere cosa accade quando l'adattatore IMS viene riavviato e quindi spiega come gestire le unità di ripristino non risolte che si verificano.

### **Cosa accade quando l'adattatore IMS viene riavviato**

Ogni volta che la connessione a IBM MQ viene riavviata, seguendo un riavvio del gestore code o un comando IMS /START SUBSYS, IMS avvia il seguente processo di risincronizzazione:

1. IMS presenta l'elenco di ID UOW (unit of work) che ritiene in dubbio per l'adattatore IBM MQ IMS una alla volta con un parametro di risoluzione Commit o Backout.
2. L'adattatore IMS trasmette la richiesta di risoluzione a IBM MQ e riporta il risultato a IMS.
3. Dopo aver elaborato tutte le richieste di risoluzione IMS, l'adattatore IMS richiama da IBM MQ un elenco di tutte le UOW IBM MQ ancora in dubbio che sono state avviate dal sistema IMS. Questi vengono notificati al terminale principale IMS nel messaggio CSQQ008I.

**Nota:** Mentre una UOW è in dubbio, tutti i messaggi IBM MQ associati vengono bloccati da IBM MQ e non sono disponibili per alcuna applicazione.

### **Come risolvere manualmente le unità di ripristino IMS**

Quando IMS si connette a IBM MQ, IBM MQ potrebbe avere una o più unità di ripristino in dubbio che non sono state risolte.

Se IBM MQ dispone di unità di ripristino in dubbio che IMS non ha risolto, viene emesso il seguente messaggio sul terminale master IMS :

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

Se viene emesso questo messaggio, IMS è stato avviato a freddo o è stato avviato con un nastro di log incompleto. Questo messaggio può essere emesso anche se IBM MQ o IMS termina in modo anomalo a causa di un errore software o di un errore di un altro sottosistema.

Dopo aver ricevuto il messaggio CSQQ008I :

- La connessione rimane attiva.
- Le applicazioni IMS possono ancora accedere alle risorse IBM MQ .
- Alcune risorse IBM MQ restano bloccate.

Se il thread in dubbio non viene risolto, è possibile avviare la creazione delle code di messaggi IMS . Se le code IMS si riempiono di capacità, IMS termina. È necessario essere consapevoli di questa potenziale difficoltà ed è necessario monitorare IMS fino a quando le unità di ripristino in dubbio non vengono completamente risolte.

### **Procedura di recupero**

Utilizzare la seguente procedura per ripristinare le unità di lavoro IMS :

1. Forzare la chiusura del log IMS, utilizzando /SWI OLDS, quindi archiviare il log IMS. Utilizzare il programma di utilità DFSERA10 per stampare i record dal nastro di log IMS precedente. I record di log di tipo X'3730' indicano una richiesta di commit phase-2 e i record di log di tipo

X'38' indicano una richiesta di interruzione. Registrare l'azione richiesta per l'ultima transazione in ciascuna regione dipendente.

2. Eseguire il lavoro batch DL/I per ripristinare ogni PSB coinvolto che non ha raggiunto un punto di commit. Il processo potrebbe richiedere del tempo perché le transazioni sono ancora in fase di elaborazione. Potrebbe anche bloccare un numero di record, che potrebbero influenzare il resto dell'elaborazione e il resto delle code messaggi.
3. Produrre un elenco delle unità di ripristino in dubbio da IBM MQ utilizzando il seguente comando:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

Viene ricevuto il seguente messaggio:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3E2C5C3F240404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-15)
UOWLOGTI(16.39.43)
UOWSTDA(2005-02-15)
UOWSTTI(16.39.43)
UOWSTATE(UNRESOLVED)
NID(IM8F .BC45A794D3810344)
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0000)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

Per IMS, l'NID è costituito dal nome della connessione IMS e da un numero univoco fornito da IMS. Il valore viene indicato in IMS come *token di recupero*. Per ulteriori informazioni, consultare *IMS Customization Guide*.

4. Confrontare gli NID (IMSID più OASN in formato esadecimale) visualizzati nei messaggi DISPLAY THREAD con gli OASN (4 byte decimali) visualizzati nell'output DFSERA10 . Decidere se eseguire il commit o il backout.
5. Eseguire la risoluzione in dubbio in IBM MQ con il comando `RESOLVE INDOUBT` , nel modo seguente:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

Per ripristinare tutti i thread associati a *nome - connessione*, utilizzare l'opzione NID (\*). Il comando restituisce uno dei seguenti messaggi per indicare se è stato eseguito il commit o il backout del thread:

```
CSQV414I THREAD network-id COMMIT SCHEDULED
CSQV415I THREAD network-id BACKOUT SCHEDULED
```

Quando si esegue una risoluzione in dubbio, IMS e l'adattatore non sono a conoscenza dei comandi per IBM MQ per eseguire il commit o il backout delle unità di ripristino in dubbio poiché solo le risorse IBM MQ sono interessate.

## **z/OS Ripristino manuale delle unità di ripristino RRS**

Utilizzare questo argomento per comprendere come determinare se vi sono unità di ripristino RRS in dubbio e come risolvere manualmente tali unità di ripristino.

Quando RRS si connette a IBM MQ, IBM MQ potrebbe avere una o più unità di ripristino in dubbio che non sono state risolte. Se IBM MQ dispone di unità di ripristino in dubbio che RRS non ha risolto, viene emesso uno dei seguenti messaggi nella console z/OS :

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Sia IBM MQ che RRS forniscono strumenti per visualizzare le informazioni sulle unità di ripristino in dubbio e le tecniche per risolverle manualmente.

In IBM MQ, utilizzare il comando DISPLAY CONN per visualizzare le informazioni sui thread IBM MQ in dubbio. L'output del comando include l'unità RRS degli ID di recupero per i thread IBM MQ che hanno RRS come coordinatore. Può essere utilizzato per determinare il risultato dell'unità di recupero.

Utilizzare il comando RESOLVE INDOUBT per risolvere manualmente il thread in dubbio IBM MQ . Questo comando può essere utilizzato per eseguire il commit o il backout dell'unità di ripristino dopo aver determinato quale sia la decisione corretta.

## **z/OS Ripristino delle unità di ripristino su un altro gestore code nel gruppo di condivisione code**

Utilizzare questo argomento per identificare e ripristinare manualmente le unità di ripristino su altri gestori code in un gruppo di condivisione code.

Se un gestore code che è un membro di un gruppo di condivisione code ha esito negativo e non può essere riavviato, altri gestori code del gruppo possono eseguire il ripristino peer e prenderne il posto. Tuttavia, il gestore code potrebbe disporre di unità di ripristino in dubbio che non possono essere risolte dal ripristino peer in quanto la disposizione finale di tale unità di ripristino è nota solo al gestore code in errore. Queste unità di ripristino vengono risolte quando il gestore code viene riavviato, ma fino a quel momento rimangono in dubbio.

Ciò significa che alcune risorse (ad esempio, i messaggi) potrebbero essere bloccate, rendendole non disponibili per altri gestori code del gruppo. In questa situazione, è possibile utilizzare il comando DISPLAY THREAD per visualizzare queste unità di lavoro sul gestore code inattivo. Se si desidera risolvere manualmente queste unità di recupero per rendere disponibili i messaggi agli altri gestori code del gruppo, è possibile utilizzare il comando RESOLVE INDOUBT.

Quando si immette il comando DISPLAY THREAD per visualizzare le unità di ripristino in dubbio, è possibile utilizzare la parola chiave QMNAME per specificare il nome del gestore code inattivo. Ad esempio, se si immette il seguente comando:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

Si riceveranno i seguenti messaggi:

```
CSQV436I +CSQ1 INDOUBT THREADS FOR QM01 -
NAME  THREAD-XREF  URID NID
USER1  0000000000000000000000000000 CSQ:0001.0
USER2  0000000000000000000000000000 CSQ:0002.0
DISPLAY THREAD REPORT COMPLETE
```

Se il gestore code specificato è attivo, IBM MQ non restituisce informazioni sui thread in dubbio, ma emette il seguente messaggio:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Utilizzare il comando IBM MQ RESOLVE INDOUBT per risolvere manualmente i thread in dubbio. Utilizzare la parola chiave QMNAME per specificare il nome del gestore code inattivo nel comando.

Questo comando può essere utilizzato per eseguire il commit o il backout dell'unità di ripristino. Il comando risolve solo la parte condivisa dell'unità di ripristino; i messaggi locali non vengono influenzati e rimangono bloccati fino a quando il gestore code non si riavvia o non si riconnette al batch CICS, IMS o RRS.

## IBM MQ e IMS

IBM MQ fornisce due componenti per l'interfaccia con IMS, l'adattatore IBM MQ - IMS e il bridge IBM MQ - IMS. Questi componenti sono comunemente denominati adattatore IMS e bridge IMS.

### Utilizzo dell'adattatore IMS

Utilizzare questo argomento per comprendere come utilizzare l'adattatore IMS, che collega IBM MQ ai sistemi IMS.

**Nota:** L'adattatore IMS non include operazioni e pannelli di controllo.

Il presente argomento contiene le seguenti sezioni:

- [“Controllo delle connessioni IMS” a pagina 523](#)
- [“Connessione dalla control region IMS” a pagina 524](#)
- [“Visualizzazione delle unità di ripristino in dubbio” a pagina 526](#)
- [“Controllo delle connessioni della regione dipendente IMS” a pagina 527](#)
- [“disconnessione daIMS” a pagina 530](#)
- [“Controllo del controllo dei trigger IMS” a pagina 530](#)

### Controllo delle connessioni IMS

Utilizzare questo argomento per comprendere i comandi dell'operatore IMS che controllano e monitorano la connessione a IBM MQ.

IMS fornisce i seguenti comandi dell'operatore per controllare e monitorare la connessione a IBM MQ:

#### **/CHANGE SUBSYS**

Elimina un'unità di ripristino in dubbio da IMS.

#### **/DISPLAY OASN SUBSYS**

Visualizza gli elementi di ripristino in sospeso.

#### **/VISUALIZZA SUBSYS**

Visualizza lo stato della connessione e l'attività del thread.

#### **/START SUBSYS**

Connette la control region IMS a un gestore code.

#### **/STOP SUBSYS**

Disconnette IMS da un gestore code.

#### **/TRACCIA**

Controlla la traccia IMS.

Per ulteriori informazioni su questi comandi, consultare il manuale *IMS/ESA Operator's Reference* per il livello di IMS che si sta utilizzando.

Le risposte del comando IMS vengono inviate al terminale da cui è stato emesso il comando. L'autorizzazione a immettere comandi IMS si basa sulla sicurezza IMS .

## **Connessione dalla control region IMS**

Utilizzare questo argomento per comprendere i meccanismi disponibili per la connessione da IMS a IBM MQ.

IMS effettua una connessione dalla propria control region a ciascun gestore code che utilizza IMS. IMS deve essere abilitato per effettuare la connessione in uno dei seguenti modi:

- Automaticamente durante:
  - Inizializzazione avvio a freddo.
  - Un avvio a caldo di IMS, se la connessione IBM MQ era attiva quando IMS è stato chiuso.
- In risposta al comando IMS :

```
/START SUBSYS sysid
```

dove *sysid* è il nome del gestore code.

Il comando può essere immesso indipendentemente dal fatto che il gestore code sia attivo o meno.

La connessione non viene effettuata fino a quando non viene effettuata la prima chiamata API MQ al gestore code. Fino a quel momento, il comando IMS /DIS SUBSYS mostra lo stato come 'NOT CONN '.

L'ordine in cui si avvia IMS e il gestore code non è significativo.

IMS non può riabilitare automaticamente la connessione al gestore code se il gestore code viene arrestato con un comando STOP QMGR, con il comando IMS /STOP SUBSYS o con una fine anomala. Pertanto, è necessario effettuare il collegamento utilizzando il comando IMS /START SUBSYS.

Se un comando IMS viene visualizzato nel log della console del gestore code in modo simile al seguente:

```
MODIFY IMS*,SS*
```

verificare il file di log principale IMS e accertarsi che IBM MQ disponga dell'autorizzazione RACF per emettere i comandi MODIFY dell'adattatore IMS .

## **Inizializzazione dell'adattatore e connessione al gestore code**

L'adattatore è una serie di moduli caricati nel controllo IMS e nelle regioni dipendenti, utilizzando la funzione di collegamento del sottosistema esterno IMS .

Questa procedura inizializza l'adattatore e si collega al gestore code:

1. Leggere il membro del sottosistema (SSM) da IMS.PROCLIB. L'SSM scelto è un parametro IMS EXEC. Esiste una voce nel membro per ogni gestore code a cui IMS può connettersi. Ogni voce contiene informazioni di controllo su un adattatore IBM MQ .
2. Caricare l'adattatore IMS .  
**Nota:** IMS carica una copia dei moduli adattatore per ogni istanza IBM MQ definita nel membro SSM.
3. Collegare l'attività del sottosistema esterno per IBM MQ.
4. Eseguire l'adattatore con il parametro CTL EXEC (IMSID) come nome connessione.

Il processo è lo stesso se la connessione fa parte dell'inizializzazione o un risultato del comando IMS / START SUBSYS.

Se il gestore code è attivo quando IMS tenta di stabilire la connessione, vengono inviati i seguenti messaggi:

- alla console z/OS :

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- al terminale master IMS :

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```

Quando IMS tenta di stabilire la connessione e *il gestore code non è attivo*, i seguenti messaggi vengono inviati al terminale master IMS ogni volta che un'applicazione effettua una chiamata MQI:

```
CSQQ001I IMS/TM imsid not connected to queue manager ssnm.  
Notify message accepted  
DFS3607I MQM1 SUBSYSTEM ID EXIT FAILURE, FC = 0286, RC = 08,  
JOBNAME = IMSEMPR1
```

Se si ricevono i messaggi DFS3607I quando si avvia la connessione a IMS o all'avvio del sistema, ciò indica che il gestore code non è disponibile. Per evitare che venga generato un numero elevato di messaggi, è necessario effettuare una delle seguenti operazioni:

1. Avviare il gestore code pertinente.
2. Immettere il comando IMS :

```
/STOP SUBSYS
```

in modo che IMS non si aspetti di connettersi al gestore code.

In caso contrario, un messaggio DFS3607I e il messaggio CSQQ001I associato vengono emessi ogni volta che un lavoro viene pianificato nella regione e ogni volta che un'applicazione esegue una richiesta di connessione al gestore code.

## Collegamento thread

In una regione MPP o IFP, IMS stabilisce una connessione thread quando il primo programma applicativo è pianificato in tale regione, anche se tale programma applicativo non effettua una chiamata IBM MQ . In una regione BMP, la connessione del thread viene effettuata quando l'applicazione effettua la prima chiamata IBM MQ ( MQCONN o MQCONNX ). Questo thread viene conservato per la durata della regione o fino a quando la connessione non viene arrestata.

Per le aree basate sui messaggi e non basate sui messaggi, l'identificativo di riferimento incrociato del thread di ripristino, *Thread-xref*, associato al thread è:

```
PSTid + PSBname
```

dove:

### **PSTid**

Identificativo area tabella specifica partizione

## PSBname

Nome blocco specifica programma

È possibile utilizzare gli ID connessione come identificativi univoci nei comandi IBM MQ , nel qual caso IBM MQ inserisce automaticamente tali ID in qualsiasi messaggio dell'operatore che genera.

## **Visualizzazione delle unità di ripristino in dubbio**

È possibile visualizzare le unità di ripristino in dubbio e tentare di recuperarle.

Le fasi operative utilizzate per elencare e ripristinare le unità di ripristino in dubbio in questo argomento sono solo per casi relativamente semplici. Se il gestore code termina in modo anomalo durante la connessione a IMS, IMS potrebbe eseguire il commit o il backout del lavoro senza che IBM MQ ne sia a conoscenza. Quando il gestore code viene riavviato, tale lavoro viene definito *in dubbio*. È necessario prendere una decisione sullo stato del lavoro.

Per visualizzare un elenco di unità di ripristino in dubbio, immettere il comando:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ risponde con un messaggio simile al seguente:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F14040404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-11-02)
UOWLOGTI(12.27.58)
UOWSTDA(2004-11-02)
UOWSTTI(12.27.58)
UOWSTATE(UNRESOLVED)
NID(CSQ1CHIN.BC0F5F1C86FC0766)
EXTURID(00000000000001F0000000007472616E5F6964547565204E6F762020...)
QMURID(000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

Per una spiegazione degli attributi in questo messaggio, consultare la descrizione del comando [DISPLAY CONN](#) .

## Ripristino delle unità di ripristino in dubbio

Per ripristinare le unità di ripristino in dubbio, immettere questo comando:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

dove:

**connection-name**

L'ID sistema IMS .

**Azione**

Indica se eseguire il commit (COMMIT) o il backout (BACKOUT) di questa unità di ripristino.

**net-node.number**

Il net-node.number associato.

Quando è stato immesso il comando RESOLVE INDOUBT, viene visualizzato uno dei seguenti messaggi:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

## Risoluzione delle voci di recupero residui

In determinati momenti, IMS crea un elenco di RRE (Residue Recovery Entries). RRE sono unità di ripristino per le quali IBM MQ potrebbe essere in dubbio. Si presentano in diverse situazioni:

- Se il gestore code non è attivo, IMS dispone di RRE che non possono essere risolti fino a quando il gestore code non è attivo. Questi RER non sono un problema.
- Se il gestore code è attivo e connesso a IMS e se IMS esegue il backout del lavoro di cui IBM MQ ha eseguito il commit, l'adattatore IMS emette il messaggio CSQQ010E. Se i dati nei due sistemi devono essere congruenti, si è verificato un problema. Per informazioni sulla risoluzione di questo problema, consultare [“Ripristino manuale delle unità di ripristino IMS” a pagina 520](#).
- Se il gestore code è attivo e connesso a IMS, potrebbero essere ancora presenti RRE anche se nessun messaggio ha informato l'utente di questo problema. Una volta stabilita la connessione IBM MQ a IMS , è possibile immettere il seguente comando IMS per scoprire se si è verificato un problema:

```
/DISPLAY OASN SUBSYS sysid
```

Per eliminare il RRE, immettere uno dei seguenti comandi IMS :

```
/CHANGE SUBSYS sysid RESET
/CHANGE SUBSYS sysid RESET OASN nnnn
```

dove *nnnn* è il numero di sequenza dell'applicazione di origine elencato in risposta al comando +CSQ1 DISPLAY. Questo è il numero di pianificazione dell'istanza del programma, che fornisce la sua posizione nella sequenza di richiami di tale programma dall'ultimo avvio a freddo di IMS . IMS non può avere due unità di ripristino in dubbio con lo stesso numero di pianificazione.

Questi comandi reimpostano lo stato di IMS ; non comportano alcuna comunicazione con IBM MQ.

## **Controllo delle connessioni della regione dipendente IMS**

È possibile controllare, monitorare e, quando necessario, terminare le connessioni tra IMS e IBM MQ.

Il controllo delle connessioni della regione dipendente IMS implica le seguenti attività:

- [Connessione dalle regioni dipendenti](#)
- [Opzioni errore regione](#)

- Monitoraggio dell'attività ... sulle connessioni
- Disconnessione dalle regioni dipendenti

## Connessione da regioni dipendenti

L'adattatore IMS utilizzato nella control region viene caricato anche nelle aree dipendenti. Viene effettuata una connessione da ciascuna regione dipendente a IBM MQ. Questa connessione è utilizzata per coordinare l'impegno del lavoro IBM MQ e IMS. Per inizializzare e stabilire la connessione, IMS effettua le seguenti operazioni:

1. Legge il membro del sottosistema (SSM) da IMS.PROCLIB.

Un membro del sottosistema può essere specificato nel parametro EXEC della regione dipendente. Se non viene specificato, viene utilizzato l'SSM della control region. Se è probabile che la regione non si colleghi mai a IBM MQ, per evitare di caricare l'adattatore, specificare un membro senza voci.

2. Carica l'adattatore IBM MQ.

Per un programma di messaggi batch, il carico non viene eseguito fino a quando l'applicazione non emette il primo comando di messaggistica. In quel momento, IMS tenta di effettuare la connessione.

Per una regione del programma di elaborazione messaggi o IMS fast - path, il tentativo viene effettuato quando la regione viene inizializzata.

## Opzioni di errore regione

Se il gestore code non è attivo o se le risorse non sono disponibili quando il primo comando di messaggistica viene inviato dai programmi applicativi, l'azione intrapresa dipende dall'opzione di errore specificata nella voce SSM. Le opzioni sono:

### **R**

Il codice di ritorno appropriato viene inviato all'applicazione.

### **Q**

L'applicazione termina in modo anomalo con codice di fine anomala U3051. Il messaggio di input viene accodato nuovamente.

### **A**

L'applicazione termina in modo anomalo con codice di fine anomala U3047. Il messaggio di input viene eliminato.

## Monitoraggio dell'attività sulle connessioni

Un thread viene stabilito da una region dipendente quando un'applicazione effettua la prima richiesta IBM MQ con esito positivo. È possibile visualizzare le informazioni sulle connessioni e le applicazioni che le utilizzano correntemente immettendo il comando seguente da IBM MQ:

```
+CSQ1 DISPLAY CONN(*) ALL
```

Il comando produce un messaggio simile al seguente:

```

CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-12-15)
UOWLOGTI(16.39.43)
UOWSTDA(2004-12-15)
UOWSTTI(16.39.43)
UOWSTATE(ACTIVE)
NID( )
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0049)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)

```

Per la control region, *thread-xref* è il valore speciale CONTROL. Per le regioni dipendenti, è il PSTid concatenato al PSBname. *auth-id* è il campo dell'utente dalla scheda di lavoro o l'ID dalla tabella delle procedure avviate z/OS .

Per una spiegazione dell'elenco visualizzato, consultare il messaggio CSQV402I nella documentazione di [IBM MQ for z/OS](#) , completamento, e codici di errore .

IMS fornisce un comando di visualizzazione per monitorare la connessione a IBM MQ. Mostra quale programma è attivo su ogni connessione della regione dipendente, il nome utente LTERM e lo stato di collegamento della regione di controllo. Il comando è:

```
/DISPLAY SUBSYS name
```

Lo stato della connessione tra IMS e IBM MQ viene visualizzato come uno tra:

```

CONNECTED
NOT CONNECTED
CONNECT IN PROGRESS
STOPPED
STOP IN PROGRESS
INVALID SUBSYSTEM NAME= name
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING

```

Lo stato del thread da ciascuna regione dipendente è uno dei seguenti:

```

CONN
CONN, ACTIVE (includes LTERM of user)

```

## Disconnessione dalle regioni dipendenti

Per modificare i valori nel membro SSM di IMS.PROCLIB, si scollega una regione dipendente. Per fare ciò, è necessario:

1. Immettere il comando IMS :

```
/STOP REGION
```

2. Aggiornare il membro SSM.
3. Immettere il comando IMS :

```
/START REGION
```

### **disconnessione daIMS**

La connessione viene terminata al termine di IMS o del gestore code. In alternativa, l'operatore del terminale master IMS può interrompere esplicitamente la connessione.

Per terminare la connessione tra IMS e IBM MQ, utilizzare il seguente comando IMS :

```
/STOP SUBSYS sysid
```

Il comando invia il seguente messaggio al terminale che lo ha emesso, di solito l'MTO (master terminal operator):

```
DFS058I STOP COMMAND IN PROGRESS
```

Il comando IMS :

```
/START SUBSYS sysid
```

è richiesto per ristabilire la connessione.

**Nota:** Il IMS comando /STOP SUBSYS non viene completato se è in esecuzione un controllo trigger IMS .

### **Controllo del controllo dei trigger IMS**

È possibile utilizzare la transazione CSQQTRMN per arrestare e avviare il controllo trigger IMS .

Il controllo trigger IMS (la transazione CSQQTRMN) è descritto in [Impostazione del controllo trigger IMS](#).

Per controllare il controllo dei trigger IMS , consultare:

- [Avvio di CSQQTRMN](#)
- [Arresto di CSQQTRMN](#)

### **Avvio di CSQQTRMN**

1. Avviare un BMP orientato al batch che esegue il programma CSQQTRMN per ogni coda di avvio che si desidera monitorare.
2. Modificare il JCL batch per aggiungere un DDname di CSQQUT1 che punti ad un dataset contenente le seguenti informazioni:

```

QMGRNAME=q_manager_name      Comment: queue manager name
INITQUEUENAME=init_q_name    Comment: initiation queue name
LTERM=lterm                  Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES         Comment: Send error messages to console

```

dove:

q_manager_name	Il nome del gestore code (se è vuoto, viene utilizzato il valore predefinito indicato in CSQQDEFV)
init_q_name	Il nome della coda di avvio da monitorare
lterm	Il nome LTERM IMS per la destinazione dei messaggi di errore (se è vuoto, il valore predefinito è MASTER).
CONSOLEMESSAGES= YES	Richiede che i messaggi inviati all'LTERM IMS designato vengano inviati anche alla console z/OS . Se questo parametro viene omissso o digitato in modo errato, il valore predefinito è NON inviare messaggi alla console.

3. Aggiungere un nome DD CSQQUT2 se si desidera un report stampato dell'elaborazione dell'input CSQQUT1 .

#### Nota:

1. Il dataset CSQQUT1 viene definito con LRECL=80. Altre informazioni DCB vengono prese dal dataset. Il DCB per il dataset CSQQUT2 è RECFM=VBA e LRECL=125.
2. È possibile inserire solo una parola chiave su ogni record. Il valore della parola chiave è delimitato dal primo spazio vuoto che segue la parola chiave; ciò significa che è possibile includere commenti. Un asterisco nella colonna 1 indica che l'intero record di input è un commento.
3. Se si utilizza una delle parole chiave QMGRNAME o LTERM , CSQQTRMN utilizza il valore predefinito per tale parola chiave.
4. Assicurarsi che il sottosistema sia avviato in IMS (mediante il comando /START SUBSYS) prima di inoltrare il lavoro BMP di controllo trigger. Se non viene avviato, il lavoro di controllo trigger termina con il codice di interruzione U3042.

## Arresto di CSQQTRMN

Una volta avviato, CSQQTRMN viene eseguito fino a quando la connessione tra IBM MQ e IMS non viene interrotta a causa di uno dei seguenti eventi:

- il gestore code in fase di chiusura
- IMS con termine in data

o viene immesso un comando z/OS STOP **jobname** .

## z/OS Controllo del bridge IMS

Utilizzare questo argomento per comprendere i comandi IMS che è possibile utilizzare per controllare il bridge IMS .

Non esistono comandi IBM MQ per controllare il bridge IBM MQ-IMS . Tuttavia, è possibile arrestare la consegna dei messaggi a IMS nei seguenti modi:

- Per le code non condivise, utilizzando il comando ALTER QLOCAL (xxx) GET (DISABLED) per tutte le code bridge.
- Per le code con cluster, utilizzando il comando SUSPEND QMGR CLUSTER (xxx). Ciò è valido solo quando un altro gestore code ospita anche la coda bridge con cluster.

- Per le code con cluster, utilizzando il comando SUSPEND QMGR FACILITY (IMSBRIDGE). Non vengono inviati ulteriori messaggi a IMS, ma le risposte per le transazioni in sospeso vengono ricevute da IMS.  
Per avviare di nuovo l'invio di messaggi a IMS , immettere il comando RESUME QMGR FACILITY (IMSBRIDGE).

È anche possibile utilizzare il comando MQSC DISPLAY SYSTEM per visualizzare se il bridge è sospeso. Consultare [Comandi MQSC](#) per dettagli su questi comandi.

Per ulteriori informazioni, consultare:

- [“Avvio e arresto del bridge IMS” a pagina 532](#)
- [“Controllo delle connessioni IMS” a pagina 532](#)
- [Controllo delle code bridge](#)
- [“Risincronizzazione del bridge IMS” a pagina 533](#)
- [Gestione dei nomi tpipe](#)
- [Eliminazione di messaggi da IMS](#)
- [Eliminazione di tpipe](#)
- [“IMS Scadenza transazione” a pagina 535](#)

## Avvio e arresto del bridge IMS

Avviare il bridge IBM MQ avviando OTMA. Utilizzare il comando IMS :

```
/START OTMA
```

o avviarlo automaticamente specificando OTMA=YES nei parametri di sistema IMS . Se OTMA è già avviato, il bridge viene avviato automaticamente al termine dell'avvio del gestore code. Un messaggio di evento IBM MQ viene prodotto all'avvio di OTMA.

Utilizzare il comando IMS :

```
/STOP OTMA
```

per arrestare la comunicazione OTMA. Quando questo comando viene immesso, viene prodotto un messaggio di evento IBM MQ .

## Controllo delle connessioni IMS

IMS fornisce questi comandi operatore per controllare e monitorare la connessione a IBM MQ:

### **/DEQUEUE TMEMBER *tmember* TPIPE *tpipe***

Rimuove i messaggi da un Tpipe. Specificare PURGE per rimuovere tutti i messaggi oppure PURGE1 per rimuovere solo il primo messaggio.

### **/VISUALIZZA OTMA**

Visualizza le informazioni di riepilogo sul server e sui client OTMA e sullo stato del client.

### **/DISPLAY TMEMBER *nome***

Visualizza le informazioni su un client OTMA.

### **/DISPLAY TRACE TMEMBER *nome***

Visualizza le informazioni su ciò che viene tracciato.

### **/SECURE OTMA**

Imposta le opzioni di sicurezza.

### **/START OTMA**

Abilita le comunicazioni tramite OTMA.

### **/START TMEMBER *tmember* TPIPE *tpipe***

Avvia il Tpipe denominato.

### **/STOP OTMA**

Arresta le comunicazioni tramite OTMA.

### **/STOP TMEMBER *tmember* TPIPE *tpipe***

Arresta il Tpipe denominato.

### **/TRACCIA**

Controlla la traccia IMS .

Per ulteriori informazioni su questi comandi, consultare il manuale *IMS/ESA Operators Reference* per il livello di IMS che si sta utilizzando.

Le risposte del comando IMS vengono inviate al terminale da cui è stato emesso il comando. L'autorizzazione a immettere comandi IMS è basata sulla sicurezza IMS .

## **Controllo delle code bridge**

Per interrompere la comunicazione con il gestore code con il nome membro XCF *tmember* tramite il bridge, immettere il seguente comando IMS :

```
/STOP TMEMBER tmember TPIPE ALL
```

Per riprendere la comunicazione, immettere il seguente comando IMS :

```
/START TMEMBER tmember TPIPE ALL
```

I Tpipe per una coda possono essere visualizzati utilizzando il comando MQ DISPLAY QUEUE.

Per arrestare la comunicazione con il gestore code su un singolo Tpipe, immettere il seguente comando IMS :

```
/STOP TMEMBER tmember TPIPE tpipe
```

Uno o due Tpipe vengono creati per ogni coda bridge attiva, pertanto l'immissione di questo comando arresta la comunicazione con la coda IBM MQ . Per riprendere la comunicazione, utilizzare il seguente comando IMS :

```
/START TMEMBER tmember TPIPE tpipe
```

In alternativa, è possibile modificare gli attributi della coda IBM MQ per renderla inibita.

## **Risincronizzazione del bridge IMS**

Il bridge IMS viene riavviato automaticamente ogni volta che il gestore code IMS o OTMA vengono riavviati.

La prima attività eseguita dal bridge IMS è la risincronizzazione con IMS. Ciò implica IBM MQ e IMS la verifica dei numeri di sequenza su ogni Tpipe sincronizzato. Un Tpipe sincronizzato viene utilizzato quando i messaggi persistenti vengono inviati a IMS da una coda bridge IBM MQ - IMS utilizzando la modalità di commit zero (commit - then - send).

Se il bridge non può risincronizzarsi con IMS, il codice di rilevamento IMS viene restituito nel messaggio CSQ2023E e la connessione a OTMA viene arrestata. Se il bridge non può risincronizzarsi con un singolo Tpipe IMS, il codice di rilevamento IMS viene restituito nel messaggio CSQ2025E e il Tpipe viene arrestato. Se un Tpipe è stato avviato a freddo, i numeri di sequenza recuperabili vengono automaticamente reimpostati su 1.

Se il bridge rileva numeri di sequenza non corrispondenti durante la risincronizzazione con un Tpipe, viene emesso il messaggio CSQ2020E. Utilizzare il comando IBM MQ RESET TPIPE per avviare la risincronizzazione con il Tpipe IMS. È necessario fornire il nome membro e il gruppo XCF e il nome del Tpipe; queste informazioni vengono fornite dal messaggio.

È anche possibile specificare:

- Un nuovo numero di sequenza recuperabile da impostare nel Tpipe per i messaggi inviati da IBM MQ e da impostare come numero di sequenza di ricezione del partner. Se non si specifica questa opzione, il numero di sequenza di ricezione del partner è impostato sul numero di sequenza di invio IBM MQ corrente.
- Un nuovo numero di sequenza recuperabile da impostare nel Tpipe per i messaggi ricevuti da IBM MQ e da impostare come numero di sequenza di invio del partner. Se non si specifica questo valore, il numero di sequenza di invio del partner è impostato sul numero di sequenza di ricezione IBM MQ corrente.

Se esiste un'unità di ripristino non risolta associata al Tpipe, questa viene notificata anche nel messaggio. Utilizzare il comando IBM MQ RESET TPIPE per specificare se eseguire il commit dell'unità di ripristino o eseguirne il backout. Se si esegue il commit dell'unità di ripristino, il batch di messaggi è già stato inviato a IMS e viene eliminato dalla coda bridge. Se si esegue il backout dell'unità di ripristino, i messaggi vengono restituiti alla coda del bridge, per essere successivamente inviati a IMS.

Modalità commit 1 (send - then - commit) I Tpipe non sono sincronizzati.

### **Considerazioni per le transazioni in modalità di commit 1**

In IMS, le transazioni in modalità di esecuzione commit 1 (CM1) inviano le relative risposte di output prima del punto di sincronizzazione.

Una transazione CM1 potrebbe non essere in grado di inviare la propria risposta, ad esempio perché:

- Il Tpipe su cui deve essere inviata la risposta è arrestato
- OTMA è arrestato
- Il client OTMA (ossia, il gestore code) è stato eliminato
- La coda di risposta e la coda di messaggi non recapitabili non sono disponibili

Per questi motivi, l'applicazione IMS che invia il messaggio pseudo - abend con codice U0119. La transazione e il programma IMS non vengono arrestati in questo caso.

Questi motivi spesso impediscono l'invio di messaggi a IMS e la consegna di risposte da IMS. Una fine anomala U0119 può verificarsi se:

- Tpipe, OTMA o il gestore code vengono arrestati mentre il messaggio si trova in IMS
- IMS risponde su un Tpipe differente al messaggio in entrata e che Tpipe è arrestato
- IMS risponde a un client OTMA differente e tale client non è disponibile.

Ogni volta che si verifica una fine anomala U0119, sia il messaggio in entrata in IMS che i messaggi di risposta in IBM MQ vengono persi. Se l'emissione di una transazione CM0 non può essere consegnata per uno qualsiasi di questi motivi, viene accodata sul Tpipe all'interno di IMS.

## Utilizzo dei nomi tpipe

Molti dei comandi utilizzati per controllare il bridge IBM MQ - IMS richiedono il nome *tpipe*. Utilizzare questo argomento per comprendere in che modo è possibile trovare ulteriori dettagli del nome *tpipe*.

Sono necessari nomi *tpipe* per molti dei comandi che controllano il bridge IBM MQ - IMS. È possibile ottenere i nomi *tpipe* dal comando DISPLAY QUEUE e prendere nota dei seguenti punti:

- I nomi *tpipe* vengono assegnati quando viene definita una coda locale
- una coda locale ha due nomi *tpipe*, uno per la sincronizzazione e uno per la non sincronizzazione
- I nomi *tpipe* non saranno noti a IMS fino a quando non avranno luogo alcune comunicazioni tra IMS e IBM MQ specifiche per quella particolare coda locale
- Perché un *tpipe* sia disponibile per l'utilizzo da parte del bridge IBM MQ - IMS, la relativa coda associata deve essere assegnata a una classe di archiviazione che abbia completati i campi del nome membro e del gruppo XCF corretti

## Eliminazione dei messaggi da IMS

Un messaggio destinato a IBM MQ tramite il bridge IMS può essere eliminato se Tmember / Tpipe viene arrestato. Per cancellare un messaggio per il gestore code con nome membro XCF *tmember*, immettere il seguente comando IMS:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

Per eliminare tutti i messaggi sul Tpipe, immettere il seguente comando IMS:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

## Eliminazione di tpipe

Non è possibile eliminare IMS *tpipes*. Vengono eliminati da IMS nei seguenti orari:

- I *tpipe* sincronizzati vengono eliminati quando IMS viene avviato a freddo.
- I *tpipe* non sincronizzati vengono eliminati quando IMS viene riavviato.

## IMS Scadenza transazione

Un'ora di scadenza è associata a una transazione; a qualsiasi messaggio IBM MQ può essere associata un'ora di scadenza. L'intervallo di scadenza viene passato dall'applicazione a IBM MQ, utilizzando MQMD.Expiry. Il tempo è la durata di un messaggio prima che scada, espressa come un valore in decimi di secondo. Un tentativo di eseguire MQGET di un messaggio, dopo che è scaduto, comporta la rimozione del messaggio dalla coda e l'esecuzione dell'elaborazione della scadenza. Il tempo di scadenza diminuisce quando un flusso di messaggi tra i gestori code su una rete IBM MQ. Quando un messaggio IMS viene passato attraverso il bridge IMS a OTMA, la restante scadenza del messaggio viene passata a OTMA come scadenza della transazione.

Se una transazione ha una scadenza specificata, OTMA fa scadere le transazioni di input in tre diverse posizioni in IMS:

- messaggio di input ricevuto da XCF
- tempo di accodamento messaggi di immissione
- tempo GU applicazione

Non viene eseguita alcuna scadenza dopo l'ora GU.

La transazione EXPRTIME può essere fornita da:

- IMS Definizione transazione
- Intestazione del messaggio OTMA IMS
- Uscita utente IMS DFSINSX0
- Comandi IMS CREATE o UPDATE TRAN

IMS indica che è scaduta una transazione terminando una transazione con 0243 e inviando un messaggio. Il messaggio emesso è DFS555I nell'ambiente di code non condivise o DFS2224I nell'ambiente di code condivise.

## z/OS **Funzionamento di Advanced Message Security su z/OS**

Lo spazio di indirizzi Advanced Message Security accetta comandi utilizzando il comando z/OS MODIFY.

Per immettere comandi per lo spazio di indirizzi Advanced Message Security (AMS), utilizzare il comando z/OS MODIFY.

Ad esempio,

```
F qmgrAMSM, cmd
```

dove *qmgr* è il prefisso del nome dell'attività avviata.

Tabella 29 a pagina 536 descrive i comandi MODIFY accettati:

Comando	Opzione	Descrizione
VISUALIZZA		Visualizza informazioni sulla versione
AGGIORNA	keyRing Politica TUTTO	Aggiornare i certificati key ring, le politiche di sicurezza o entrambi.
SMFAUDIT	RIUSCITO NON RIUSCITO TUTTO	Impostare se il controllo SMF è richiesto quando AMS protegge o annulla correttamente la protezione dei messaggi, quando AMS non riesce a proteggere o annullare la protezione dei messaggi o entrambi.
PE SMFT	0 - 255	Impostare il tipo di record SMF da generare quando AMS protegge o non protegge i messaggi. Per disabilitare il controllo SMF specificare un record di tipo 0.

**Nota:** Per specificare un'opzione, è necessario che sia separata da una virgola. Ad esempio:

```
F qmgrAMSM, REFRESH KEYRING  
F qmgrAMSM, SMFAUDIT ALL  
F qmgrAMSM, SMFTYPE 180
```

### Aggiorna comando

Le modifiche rese effettive dall'immissione del comando **REFRESH** si applicano alle applicazioni che emettono MQOPEN dopo che il comando **REFRESH** è stato completato. Le applicazioni esistenti che hanno

una coda aperta, continuano ad utilizzare le opzioni da quando l'applicazione ha aperto la coda. Per utilizzare i nuovi valori, l'applicazione deve chiudere e riaprire la coda.

## avvio e arrestoAMS

Non è necessario immettere un comando per avviare o arrestare lo spazio di indirizzo Advanced Message Security . Lo spazio di indirizzo AMS viene avviato automaticamente quando il gestore code viene avviato se AMS è stato abilitato con il parametro **SPLCAP** di CSQ6SYSPE viene arrestato quando il gestore code viene arrestato.

## AmministrazioneIBM MQ Internet Pass-Thru

---

Questa sezione descrive come amministrare IBM MQ Internet Pass-Thru (MQIPT).

Configurare MQIPT apportando le modifiche al file di configurazione `mqipt.conf` come descritto in [Configurazione di IBM MQ Internet Pass-Thru](#). Per amministrare MQIPT, incluso l'aggiornamento di MQIPT per rendere effettive le modifiche alla configurazione senza riavviare MQIPT, utilizzare il comando **mqiptAdmin** . Per informazioni sulla gestione di MQIPT utilizzando il comando **mqiptAdmin** , consultare [“Gestione di MQIPT utilizzando la riga comandi” a pagina 539](#).

## avvio e arrestoMQIPT

È possibile avviare MQIPT dalla riga comandi o avviarlo automaticamente quando il sistema viene avviato. È possibile arrestare MQIPT utilizzando il comando **mqiptAdmin** .

### Avvio di MQIPT dalla riga comandi

MQIPT è installato in una directory di installazione, ad esempio:

-  **Windows** C : \MQIPT su sistemi Windows , con script eseguibili in C : \MQIPT\bin
-  **Linux**  **AIX** /opt/mqipt su sistemi AIX and Linux , con script eseguibili in /opt/mqipt/bin

MQIPT utilizza anche una directory home, che contiene il file di configurazione `mqipt.conf` e tutti i file emessi da MQIPT quando è in esecuzione. Le seguenti sottodirectory della directory home MQIPT vengono create automaticamente quando MQIPT viene richiamato per la prima volta:

- Una directory `errors` in cui vengono scritti i file di traccia e First Failure Support Technology (FFST)
- Una directory `logs` in cui viene conservato il log di connessione

L'ID utente con cui viene eseguito MQIPT deve disporre dell'autorizzazione per creare tali directory oppure, in alternativa, le directory devono già esistere e l'ID utente deve disporre dell'autorizzazione per creare, leggere e scrivere file in esse. Inoltre, se si utilizza una politica Java security manager , la politica di sicurezza deve concedere le autorizzazioni richieste per queste directory. Per ulteriori informazioni sulle impostazioni della politica di Security Manager, fare riferimento a [Java security manager](#).

È possibile utilizzare la directory di installazione come directory home. Se si utilizza questa directory, è necessario assicurarsi che l'ID utente con cui viene eseguito MQIPT disponga delle autorizzazioni appropriate e che tutte le politiche di Security Manager siano configurate correttamente.

Per avviare MQIPT, utilizzare il comando **mqipt** , che si trova nella directory `bin` della directory di installazione MQIPT . Ad esempio, il seguente comando avvia un'istanza di MQIPT che utilizza la directory `C:\mqiptHome` come directory home:

```
mqipt C:\mqiptHome
```

Per ulteriori informazioni sul comando **mqipt** , consultare [mqipt \(start MQIPT\)](#).

 **V 9.2.0**  **V 9.2.0** È possibile utilizzare il comando **mqipt** per specificare un nome da assegnare all'istanza MQIPT in fase di avvio. Il nome dell'istanza MQIPT viene utilizzato per gestire le istanze locali di

MQIPT con il comando **mqiptAdmin** senza dover utilizzare una porta comandi. Se questo parametro non viene specificato, il nome della directory home MQIPT viene utilizzato come nome dell'istanza MQIPT .

I messaggi della console mostrano lo stato di MQIPT. Se si verifica un errore, consultare [Risoluzione dei problemi IBM MQ Internet Pass-Thru](#). I seguenti messaggi sono un esempio dell'output quando MQIPT viene avviato correttamente:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is C:\mqiptHome
MQCPI021 Password checking has been enabled on the command port
MQCPI144 MQ Advanced capabilities not enabled
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1414 is starting and will forward messages to :
MQCPI034 ...examplehost(1414)
MQCPI035 ...using MQ protocols
MQCPI057 ...trace level 5 enabled
MQCPI078 Route 1414 ready for connection requests
```

## Avvio automatico di MQIPT

È possibile installare MQIPT come servizio di sistema che viene avviato automaticamente all'avvio del sistema. Utilizzare il comando **mqiptService** per installare e disinstallare il servizio MQIPT .

- **Windows** Sui sistemi Windows , il comando **mqiptService** installa MQIPT come servizio Windows .
- **Linux** **AIX** Sui sistemi AIX and Linux , il comando **mqiptService** installa MQIPT come servizio init System V che viene avviato all'avvio del sistema. Su sistemi di Linux che non supportano System V init, utilizzare un altro metodo, come systemd, per gestire MQIPT come un servizio.

Quando il servizio MQIPT viene avviato, vengono avviati tutti gli instradamenti MQIPT attivi. Quando il servizio viene arrestato, tutti i percorsi vengono immediatamente arrestati.

È possibile installare solo un servizio MQIPT su un sistema, anche se sul sistema è presente più di una installazione di MQIPT .

Per ulteriori informazioni relative al comando **mqiptService** , consultare [mqiptService \(gestione del servizio MQIPT\)](#) .

## In fase di arrestoMQIPT

È possibile arrestare MQIPT utilizzando il comando **mqiptAdmin** con il parametro **-stop** .

**V9.2.0** **V9.2.0** Ad esempio, il comando riportato di seguito arresta un'istanza di MQIPT con il nome **mqipt1** in esecuzione localmente con lo stesso ID utente del comando **mqiptAdmin** :

```
mqiptAdmin -stop -n ipt1
```

Il comando **mqiptAdmin** si connette all'istanza attiva di MQIPT da gestire utilizzando uno dei seguenti metodi:

- **V9.2.0** **V9.2.0** collegandosi a un'istanza locale di MQIPT senza utilizzare la porta comandi.
- creando una connessione di rete a una porta comandi.

L'arresto remoto deve essere abilitato impostando la proprietà **RemoteShutDown** su **true** prima che il comando **mqiptAdmin** possa essere utilizzato per arrestare MQIPT inviando un comando a una porta comandi.

Per ulteriori informazioni sulla gestione di MQIPT mediante l'utilizzo del comando **mqiptAdmin** , consultare [“Gestione di MQIPT utilizzando la riga comandi”](#) a pagina 539.

Da IBM MQ 9.1.5, se la configurazione MQIPT contiene password codificate utilizzando una chiave di codifica diversa dalla chiave predefinita, è necessario fornire la chiave di codifica password in un file che MQIPT può leggere all'avvio.

### **Il file della chiave di crittografia della parola d'ordine**

Le password codificate per essere memorizzate e utilizzate da MQIPT possono essere codificate utilizzando una chiave di cifratura fornita dall'utente. Se non si fornisce una chiave di codifica, viene utilizzata la chiave di codifica predefinita. Non è necessario specificare una chiave di codifica della password, tuttavia è più sicuro farlo. Se non si specifica la propria chiave di codifica, viene utilizzata la chiave di codifica predefinita.

Se si fornisce una chiave di crittografia della password, è necessario memorizzarla in un file a cui è possibile accedere tramite il comando **mqiptPW** utilizzato per crittografare le password e MQIPT. Le uniche restrizioni sul contenuto del file sono che deve contenere almeno un carattere e una sola riga di testo.

**Nota:** È necessario assicurarsi che le autorizzazioni file appropriate siano impostate sul file di chiavi di codifica password per impedire agli utenti non autorizzati di leggere la chiave di codifica. Solo l'utente che esegue il comando **mqiptPW** e l'utente con cui viene eseguito MQIPT necessitano dell'autorità per leggere la chiave di codifica della parola d'ordine.

La stessa chiave di codifica della password viene utilizzata per codificare e decodificare tutte le password memorizzate per un'istanza di MQIPT. Pertanto, è necessario un solo file della chiave di codifica della password per ogni installazione di MQIPT .

Se la chiave di crittografia della password per un'installazione MQIPT viene modificata, tutte le password crittografate devono essere crittografate nuovamente utilizzando la nuova chiave di crittografia.

### **In fase di avvioMQIPT**

Il nome predefinito del file della chiave di codifica della password è *MQIPT\_HOME\_DIR/mqipt\_cred.key*, dove *MQIPT\_HOME\_DIR* è la directory in cui è memorizzato il file di configurazione *mqipt.conf* . Se si intende eseguire MQIPT come un servizio che viene avviato automaticamente, è necessario creare il file di chiavi di codifica della password con il nome predefinito.

Se il file della chiave di codifica della password viene creato con un nome diverso dal nome predefinito, il nome del file deve essere fornito a MQIPT quando viene avviato. Il nome del file della chiave di crittografia della password può essere specificato utilizzando uno dei seguenti metodi, in ordine di preferenza:

1. il parametro **-sf** del comando **mqipt** utilizzato per avviare MQIPT.
2. la variabile di ambiente *MQS\_MQIPTCRED\_KEYFILE* .
3. la proprietà di *com.ibm.mq.ipc.cred.keyfile* Java .

Se non viene fornito alcun nome file della chiave di codifica della password, verrà utilizzato il nome file predefinito, se il file esiste. Se il file della chiave di codifica della password predefinita non esiste, viene utilizzata la chiave di codifica della password predefinita.

## **Gestione di MQIPT utilizzando la riga comandi**

È possibile utilizzare il comando **mqiptAdmin** sulla riga comandi per gestire MQIPT.

È possibile utilizzare il comando **mqiptAdmin** per eseguire le seguenti funzioni di gestione:

- Elencare le istanze locali attive di MQIPT.
- Aggiornare un'istanza di MQIPT dopo aver apportato modifiche al file di configurazione.
- Arrestare un'istanza di MQIPT.

Il comando `mqiptAdmin` si trova nella sottodirectory `bin` della directory di installazione MQIPT .

Il comando **`mqiptAdmin`** si connette all'istanza attiva di MQIPT da gestire utilizzando uno dei seguenti metodi:

- creando una connessione di rete a una porta comandi.
-   collegandosi a un'istanza locale di MQIPT senza utilizzare la porta comandi.

Il comando **`mqiptAdmin`** è compatibile con le versioni precedenti di MQIPT, ma non è possibile utilizzare il comando per gestire le versioni di MQIPT che sono una versione successiva rispetto alla versione del comando **`mqiptAdmin`** . In un ambiente che include diverse versioni di MQIPT, è necessario utilizzare la versione più recente del comando **`mqiptAdmin`** .

Per ulteriori informazioni relative alla sintassi del comando **`mqiptAdmin`** , consultare [mqiptAdmin](#) (amministrare MQIPT).

## Gestione locale senza una porta comandi



Da IBM MQ 9.2.0, le istanze locali di MQIPT possono essere gestite senza utilizzare una porta comandi. L'amministrazione locale consente di gestire MQIPT utilizzando il comando **`mqiptAdmin`** solo quando viene eseguita nello stesso sistema dell'istanza MQIPT che si desidera amministrare.

Affinché **`mqiptAdmin`** sia autorizzato a gestire un'istanza locale di MQIPT senza utilizzare la porta comandi, l'istanza MQIPT deve essere in esecuzione sullo stesso sistema e con lo stesso ID utente di **`mqiptAdmin`**. In alternativa, su AIX and Linux, **`mqiptAdmin`** può essere eseguito come `root`.

L'amministrazione locale è abilitata per impostazione predefinita. Per disattivare l'amministrazione locale, utilizzare la proprietà di configurazione **`LocalAdmin`** . Per ulteriori informazioni sulla proprietà **`LocalAdmin`** , consultare [LocalAdmin](#).

Per amministrare le istanze locali di MQIPT, è necessario assegnare un nome a ciascuna istanza. È possibile assegnare un nome a un'istanza di MQIPT utilizzando il parametro **`-n`** quando si avvia MQIPT con il comando **`mqipt`** . Se non si specifica un nome all'avvio di MQIPT, il nome della directory home viene utilizzato come nome dell'istanza MQIPT . Ad esempio, il seguente comando avvia MQIPT e assegna il nome `ipt1` all'istanza:

```
mqipt /opt/mqipt1 -n ipt1
```

Una volta che l'istanza ha un nome, è possibile gestire tale istanza specificando il nome nel comando **`mqiptAdmin`** con il parametro **`-n`** . Ad esempio, il seguente comando arresta l'istanza locale di MQIPT con nome `ipt1`:

```
mqiptAdmin -stop -n ipt1
```

È possibile visualizzare tutte le istanze attive locali di MQIPT che il comando **`mqiptAdmin`** è autorizzato a gestire senza utilizzare una porta comandi utilizzando il comando **`mqiptAdmin`** con il parametro **`-list`** . Ad esempio, il seguente comando elenca tutte le istanze attive locali di MQIPT che l'utente che ha avviato il comando **`mqiptAdmin`** è autorizzato a gestire:

```
mqiptAdmin -list
```

## Gestione mediante una porta comandi



Da IBM MQ 9.2.0, è possibile configurare MQIPT con una porta comandi non protetta e una porta comandi protetta con TLS. È possibile utilizzare queste porte comandi per gestire MQIPT come qualsiasi utente che si trova sullo stesso sistema dell'istanza MQIPT che si desidera gestire o da un sistema remoto.

Le versioni precedenti di MQIPT accettarono solo i comandi di gestione emessi sulla porta comandi non protetta.

**Nota:** Le connessioni alla porta comandi non protetta non vengono codificate, quindi i dati inviati sulla rete alla porta comandi non protetta, inclusa la password di accesso MQIPT , possono essere visibili ad altri utenti sulla rete.

Affinché MQIPT sia in ascolto su una porta comandi per i comandi emessi dal comando **mqiptAdmin** , è necessario specificare un valore per le proprietà **CommandPort** o **SSLCommandPort** nella sezione globale del file di configurazione `mqipt.conf` .

Esaminare le considerazioni sulla sicurezza in [Altre considerazioni sulla sicurezza](#) prima di abilitare una delle porte di comandi MQIPT . Abilitare l'autenticazione per i comandi ricevuti dalle porte dei comandi. Per ulteriori informazioni sull'autenticazione della porta comandi, consultare [“Autenticazione porta comandi”](#) a pagina 544.

Per gestire un'istanza di MQIPT utilizzando una porta comandi, specificare l'indirizzo di rete dell'host su cui è in esecuzione MQIPT e il numero di porta comandi, come parametri per il comando **mqiptAdmin** . Ad esempio, per aggiornare l'istanza MQIPT in esecuzione su `mqipt.example.com` come con la porta comandi non protetta configurata per l'ascolto sulla porta 1890, immettere il seguente comando:

```
mqiptAdmin -refresh -r mqipt.example.com:1890
```

Se non si specifica il nome host e il numero di porta, **mqiptAdmin** tenta di collegarsi a localhost, porta 1881.

Per ulteriori informazioni sulla gestione di MQIPT utilizzando la porta comandi TLS, consultare [“Amministrare di MQIPT utilizzando la porta comandi TLS”](#) a pagina 541.

## Amministrazione di MQIPT utilizzando la porta comandi TLS

Da IBM MQ 9.2.0, MQIPT può essere configurato per utilizzare una porta comandi TLS per ascoltare i comandi di gestione emessi dal comando **mqiptAdmin** . L'uso della porta comandi TLS protegge i dati sensibili come la password di accesso MQIPT sulla rete tra **mqiptAdmin** e MQIPT. Utilizzare questa procedura per configurare la porta comandi TLS e gestire MQIPT utilizzando la porta comandi TLS.

### Informazioni su questa attività

La porta comandi TLS deve essere configurata con un certificato server memorizzato in un keyring PKCS #12 o in un hardware crittografico che supporta PKCS #11 Cryptographic Token Interface. Il certificato del server della porta comandi viene inviato al comando **mqiptAdmin** durante l'handshake TLS. Questa attività presuppone che si richieda un nuovo certificato server da una CA (Certificate Authority) attendibile e che il certificato venga restituito all'utente in un file. Il comando **mqiptAdmin** convalida il certificato della porta comandi utilizzando il certificato CA della CA che ha firmato il certificato server. Il certificato CA deve essere memorizzato in un keyring PKCS #12 a cui il comando **mqiptAdmin** può accedere.

L'autenticazione del certificato del client non è supportata dalla porta comandi TLS. Per abilitare l'autenticazione per i comandi di gestione emessi su una porta comandi, vedere [“Autenticazione porta comandi”](#) a pagina 544.

Questa procedura descrive come gestire i keyring e i certificati digitali richiesti per utilizzare la porta comandi TLS utilizzando la CLI (command - line interface) **mqiptKeycmd** (iKeyman). Puoi utilizzare la CLI utilizzando il comando **mqiptKeycmd** . Per ulteriori informazioni su altri comandi che possono essere utilizzati per gestire i key ring e i certificati digitali, consultare [mqiptKeyman](#) e [mqiptKeycmd](#).

### Procedura

1. Seguire questa procedura per configurare la porta comandi TLS per l'istanza di MQIPT.
  - a) Creare un file keyring PKCS #12 da utilizzare con la porta comandi TLS. Il keyring viene utilizzato per memorizzare il certificato del server della porta comandi TLS.  
Per creare un file key ring utilizzando la CLI, immettere il seguente comando:

```
mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
```

dove *nomefile* è il nome del file key ring da creare e *password* è la password key ring.

- b) Creare una richiesta di certificato per il certificato del server della porta comandi TLS firmato dalla CA.

Per creare una richiesta di certificato utilizzando la CLI iKeyman , immetti il seguente comando:

```
mqiptKeycmd -certreq -create -db filename -pw password  
-label label -size key_size -sig_alg algorithm  
-dn distinguished_name -file certreq_filename -type pkcs12
```

dove:

**-db *nomefile***

Specifica il nome del file key ring.

**-pw *password***

Specifica la parola d'ordine del key ring.

**-label *label***

Specifica l'etichetta del certificato.

**-size *dimensione\_chiave***

Specifica la dimensione della chiave.

**-sig\_alg *algoritmo***

Specifica l'algoritmo di firma asimmetrico utilizzato per creare la coppia di chiavi della voce.

**-dn *nome\_distinto***

Specifica il nome distinto X.500 racchiuso tra virgolette.

**-file *nomefile\_certreq***

Specifica il nome file per la richiesta di certificato.

- c) Inviare il file di richiesta di certificato creato nel passo [“1.b” a pagina 542](#) alla propria CA da firmare.

- d) Dopo che la CA ha inviato il certificato firmato, riceverlo nel file key ring.

Per ricevere il certificato firmato nel key ring utilizzando la CLI, immettere il seguente comando:

```
mqiptKeycmd -cert -receive -file cert_filename -db filename  
-pw password -type pkcs12
```

dove *cert\_filename* è il nome del file contenente il certificato, *filename* è il nome del file key ring e *password* è la password key ring.

- e) Codificare la parola d'ordine key ring utilizzando il comando **mqiptPW** .

Immettere il seguente comando:

```
mqiptPW -sf encryption_key_file
```

dove *encryption\_key\_file* è il nome di un file che contiene la chiave di codifica della password per l'installazione di MQIPT . Non è necessario specificare il parametro **-sf** se l'installazione di MQIPT utilizza la chiave di codifica della password predefinita. Immettere la password key ring da codificare quando richiesto.

Per ulteriori informazioni sul comando **mqiptPW** , consultare [Codifica di una password key ring](#).

- f) Modificare il file di configurazione `mqipt.conf` e specificare le seguenti proprietà per configurare la porta comandi TLS:

i) Impostare il valore della proprietà **SSLCommandPort** sul numero di porta del comando TLS.

ii) Impostare il valore della proprietà **SSLCommandPortKeyRing** sul nome file del file di chiavi creato nel passo [“1.a” a pagina 541](#).

- iii) Impostare il valore di **SSLCommandPortKeyRingPW** sull'output della stringa mediante il comando **mqiptPW** nel passo [“1.e” a pagina 542](#).

- iv) Impostare il valore della proprietà **SSLCommandPortSiteLabel** sul nome etichetta del certificato della porta comandi TLS, specificato quando si crea la richiesta di certificato nel passo “1.b” a pagina 542.
- v) Se si desidera limitare le connessioni in entrata alla porta comandi TLS a quelle provenienti da una particolare interfaccia di rete, impostare il valore della proprietà **SSLCommandPortListenerAddress** su un indirizzo di rete appartenente a una delle interfacce di rete sul sistema su cui è in esecuzione MQIPT . Ad esempio, per limitare le connessioni in entrata alla porta comandi TLS solo a quelle provenienti dalla macchina locale, impostare il valore della proprietà **SSLCommandPortListenerAddress** su localhost.
- g) Avviare o aggiornare MQIPT per abilitare la porta comandi TLS.  
MQIPT emette messaggi della console come quelli riportati di seguito per visualizzare la configurazione della porta comandi TLS in vigore:

```
MQCPI155 Listening for control commands on port 1882 on local address * using TLS
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file c:\\iptHome\\ssl\\commandport.p12
MQCPI072 .....and certificate label mqiptadmin
```

2. Sul sistema in cui il comando **mqiptAdmin** viene utilizzato per amministrare MQIPT, attenersi alla seguente procedura per consentire a **mqiptAdmin** di connettersi alla porta comandi TLS.

- a) Creare un key ring PKCS #12 da utilizzare come truststore dal comando **mqiptAdmin** .

Per creare un file key ring utilizzando la CLI, immettere il seguente comando:

```
mqiptKeycmd -keydb -create -db filename -pw password -type pkcs12
```

dove *nomefile* è il nome del file key ring da creare e *password* è la password key ring.

- b) Importare il certificato CA della CA che ha firmato il certificato della porta comandi TLS nel keyring creato nel passaggio “2.a” a pagina 543.

Per importare il certificato CA utilizzando la CLI iKeyman , immetti il seguente comando:

```
mqiptKeycmd -cert -add -db filename -pw password -type pkcs12
-label certlabel -file cert_filename
```

dove:

**nome file**

Specifica il nome del file key ring

**Password**

Specifica la parola d'ordine key ring

**etichetta certificato**

Specifica l'etichetta da fornire al certificato CA

**nomefile\_cert**

Specifica il nome del file contenente il certificato CA

- c) Codificare la parola d'ordine key ring utilizzando il comando **mqiptPW** .

Immettere il seguente comando:

```
mqiptPW -sf encryption_key_file
```

dove *encryption\_key\_file* è il nome del file che contiene la chiave di codifica della password. Il file di chiavi di codifica password può essere diverso da quello utilizzato per codificare le password nella configurazione MQIPT . La chiave di codifica della password predefinita viene utilizzata se non si specifica un file della chiave di codifica con il parametro **-sf** . Immettere la password key ring da codificare quando richiesto.

Per ulteriori informazioni sul comando **mqiptPW** , consultare [Codifica di una password key ring](#).

- d) Creare un file delle proprietà che deve essere utilizzato dal comando **mqiptAdmin** e specificare le seguenti proprietà:

```
SSLClientCAKeyRing=key_ring_file_name
SSLClientCAKeyRingPW=key_ring_password
PasswordProtectionKeyFile=encryption_key_file
```

dove:

**nome\_file\_ring\_chiavi**

è il nome del file di chiavi creato nella fase [“2.a”](#) a pagina 543.

**password\_ring\_chiave**

è l'emissione della password codificata dal comando **mqiPTPW** nel passo [“2.c”](#) a pagina 543.

**ENCRYPTION\_KEY\_FILE**

è il nome del file che contiene la chiave di crittografia della password. È necessario specificare la proprietà **PasswordProtectionKeyFile** solo se è stato utilizzato un file di chiavi di codifica per codificare la password del key ring nel passo [“2.c”](#) a pagina 543.

- e) Immettere il comando **mqiPTAdmin** per gestire MQIPT, specificando il parametro **-s** per indicare che è richiesta una connessione TLS e il parametro **-p** per specificare il nome del file delle proprietà creato nel passo [“2.d”](#) a pagina 543.

Ad esempio, immetti il seguente comando per aggiornare un'istanza di MQIPT inviando un comando di aggiornamento alla porta comandi TLS:

```
mqiPTAdmin -refresh -r hostname:port -s -p properties_file
```

Il comando **mqiPTAdmin** emette un messaggio come il seguente per confermare che la connessione a MQIPT è protetta con TLS:

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

## Operazioni successive

Per abilitare l'autenticazione per i comandi ricevuti dalla porta comandi TLS, seguire la procedura in [“Autenticazione porta comandi”](#) a pagina 544.

## V 9.2.0 V 9.2.0 Autenticazione porta comandi

MQIPT può essere configurato per autenticare i comandi ricevuti dalla porta comandi non protetta e dalla porta comandi TLS utilizzando una password. Utilizzare questa procedura per abilitare l'autenticazione della porta comandi.

## Informazioni su questa attività

Il comando **mqiPTAdmin** richiede agli utenti di immettere una parola d'ordine quando il comando si connette alla porta comandi di una istanza di MQIPT che ha l'autenticazione della porta comandi abilitata. MQIPT convalida la parola d'ordine immessa nel comando **mqiPTAdmin** rispetto alla parola d'ordine di accesso specificata nella configurazione di MQIPT .

Le proprietà impostate per l'autenticazione della porta comandi si applicano alla porta comandi TLS e alla porta comandi non protetta.

## Procedura

1. Codificare la password di accesso MQIPT utilizzando il comando **mqiPTPW** .

Immettere il seguente comando:

```
mqiPTPW -sf encryption_key_file
```

dove *encryption\_key\_file* è il nome del file che contiene la chiave di codifica della password per l'installazione di MQIPT . Non è necessario specificare il parametro **-sf** se l'installazione di MQIPT utilizza la chiave di codifica della password predefinita. Immettere la password di accesso da codificare quando richiesto.

Per ulteriori informazioni sulla codifica delle password nella configurazione di MQIPT , vedi [Codifica delle password memorizzate](#).

2. Modificare il file di configurazione `mqipt.conf` e specificare le seguenti proprietà:

```
AccessPW=encrypted_password
RemoteCommandAuthentication=auth_setting
```

dove:

#### **password\_codificata**

è l'emissione della password codificata dal comando **mqiptPW** nel passo [“1” a pagina 544](#).

#### **auth\_setting**

è il requisito di autenticazione. L'autenticazione della porta comandi è abilitata se questa proprietà è impostata su uno dei seguenti valori:

##### **facoltativo**

Una password non è richiesta, ma se viene fornita deve essere valida. Questa opzione potrebbe essere utile durante la migrazione, ad esempio.

##### **richiesto**

È necessario fornire una password valida con ogni comando ricevuto da una porta comandi.

Per ulteriori informazioni su queste proprietà, consultare [MQIPT proprietà globali](#).

3. Avviare o aggiornare MQIPT per rendere effettive le modifiche.

MQIPT emette un messaggio che indica se l'autenticazione della porta comandi è abilitata. Ad esempio, se MQIPT è configurato per richiedere l'immissione di una password valida ogni volta che viene eseguito il comando **mqiptAdmin** , viene emesso il seguente messaggio:

```
MQCPI021 Password checking has been enabled on the command port
```

## Esecuzione di backup

È necessario eseguire il backup di alcuni file MQIPT come parte delle normali procedure di backup.

Eseguire regolarmente il backup dei seguenti file:

- Il file di configurazione, `mqipt.conf`
- I file keyring SSL/TLS specificati dalle seguenti proprietà in `mqipt.conf`:
  - **SSLClientKeyRing**
  - **SSLClientCAKeyRing**
  - **SSLServerKeyRing**
  - **SSLServerCAKeyRing**
  - **V9.2.0 V9.2.0 SSLCommandPortKeyRing**
- I file di password keyring SSL/TLS specificati dalle seguenti proprietà in `mqipt.conf`:
  - **SSLClientKeyRingPW**
  - **SSLClientCAKeyRingPW**
  - **SSLServerKeyRingPW**
  - **SSLServerCAKeyRingPW**
- **V9.2.0** Il file della chiave di codifica della password, se la configurazione MQIPT contiene password codificate con una chiave di codifica diversa dalla chiave predefinita.
- Il file della politica specificato da **SecurityManagerPolicy**, se tale proprietà è stata impostata.
- I file di uscita di sicurezza e i file di uscita del certificato specificati dalle proprietà riportate di seguito in `mqipt.conf`:
  - **SecurityExitName**

– **SSLExitName**

- I file di log di connessione nella sottodirectory `log` della directory home MQIPT , se sono necessari per scopi di controllo.

## Ottimizzazione delle prestazioni

È possibile ottimizzare le prestazioni relative di ciascun instradamento MQIPT utilizzando una combinazione di un pool di thread e una specifica di timeout di inattività.

### Thread di connessione

A ciascun instradamento MQIPT viene assegnato un lotto di lavoro di thread in esecuzione simultanea che gestiscono richieste di comunicazione in entrata. All'inizializzazione, viene creato un lotto di thread (della dimensione specificata nell'attributo `MinConnectionThreads` dell'instradamento) e viene assegnato un thread per gestire la prima richiesta in entrata. Quando arriva questa richiesta, viene assegnato un altro thread, pronto per la successiva richiesta in entrata. Quando tutti i thread vengono assegnati per il lavoro, viene creato un nuovo thread, aggiunto al lotto di lavoro e assegnato per il lavoro.

In questo modo, il pool cresce fino a quando non viene raggiunto il numero massimo di thread (specificato in **MaxConnectionThreads**). I thread vengono rilasciati di nuovo nel lotto quando termina una conversazione o quando è trascorso il periodo di timeout di inattività specificato. Quando viene raggiunto il numero massimo di sottoprocessi di lavoro, la successiva richiesta in entrata attende che un sottoprocesso venga rilasciato nuovamente al lotto di lavoro.

È possibile ridurre il tempo di attesa delle richieste aumentando il numero di thread disponibili. Tuttavia, è necessario bilanciare questo aumento con le risorse di sistema disponibili.

### Timeout di inattività

Per impostazione predefinita, i thread di lavoro non vengono terminati a causa di inattività. Quando un thread è stato assegnato a una conversazione, rimane assegnato a tale conversazione fino a quando non viene chiuso normalmente, l'instradamento viene disattivato o MQIPT viene arrestato. Facoltativamente, è possibile specificare un intervallo di timeout di inattività (in minuti) nella proprietà **IdleTimeout** in modo che i thread che sono stati inattivi per il periodo di tempo specificato vengano riciclati. I thread vengono riciclati per essere utilizzati rimettendoli nel pool di lavoro.

Se l'attività IBM MQ è intermittente, impostare il relativo intervallo di heartbeat su un valore inferiore a quello del timeout MQIPT in modo che i thread non vengano costantemente riciclati.

## Informazioni particolari

---

Queste informazioni sono state sviluppate per i prodotti ed i 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. È comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri programmi e/o prodotti, fatta eccezione per quelli espressamente indicati dall'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:

Director of Commercial Relations  
IBM Corporation  
Schoenaicher Str. 220  
D-7030 Boeblingen  
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  
Legge sulla proprietà intellettuale e legale  
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.

Qualsiasi riferimento a siti Web non IBM contenuto nelle presenti informazioni è fornito per consultazione e non vuole in alcun modo promuovere i suddetti siti Web. I materiali presenti in tali siti Web non sono parte dei materiali per questo prodotto IBM e l'utilizzo di tali siti Web è a proprio rischio.

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 termini di 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 illustrarle nel modo più completo possibile, gli esempi includono i nomi di individui, 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 WebSphere 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 perché 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 e/o in altri paesi.

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: