

9.4

Configurazione di IBM MQ

IBM

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in [“Informazioni particolari” a pagina 1063](#).

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

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

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

Indice

Configurazione.....	7
Creazione di gestori code su più piattaforme.....	7
Directory effimera configurabile.....	10
Directory dati utente.....	11
Creazione di un gestore code predefinito.....	12
Impostazione di un gestore code esistente come predefinito.....	13
Backup dei file di configurazione dopo la creazione di un gestore code.....	14
Spostamento di un gestore code su un sistema operativo differente.....	15
Configurazione delle connessioni tra client e server.....	16
Il tipo di comunicazione da utilizzare.....	17
Come configurare un IBM MQ MQI client.....	19
Configurazione di un client transazionale esteso.....	20
Definizione di canali MQI.....	30
Creazione e utilizzo di canali AMQP.....	31
Creazione di definizioni di connessione server e di connessione client su piattaforme differenti....	36
Creazione di definizioni di connessioni server e client sul server.....	42
Programmi di uscita canale per canali MQI.....	59
Connecting a client to a queue sharing group.....	63
Utilizzo delle variabili d'ambiente IBM MQ.....	64
Descrizioni delle variabili di ambiente.....	66
Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms.....	93
File di configurazione IBM MQ , mqs.ini.....	94
File di configurazione del gestore code, qm.ini.....	107
File di configurazione dell'installazione, mqinst.ini.....	167
IBM MQ MQI client file di configurazione, mqclient.ini.....	168
File di configurazione della traccia di attività, mqat.ini.....	202
Configurazione dell'accodamento distribuito.....	205
Tecniche di accodamento distribuito IBM MQ.....	206
Introduzione alla gestione delle code distribuite.....	226
Monitoraggio e controllo dei canali su AIX, Linux, and Windows.....	258
Monitoraggio e controllo dei canali su IBM i.....	282
Configurazione di un cluster di gestore code.....	303
Configurazione di un cluster uniforme.....	425
Configurazione della messaggistica di pubblicazione/sottoscrizione.....	448
Impostazione degli attributi dei messaggi di pubblicazione / sottoscrizione accodati.....	448
Avvio della pubblicazione / sottoscrizione accodata.....	450
Arresto della pubblicazione / sottoscrizione in coda.....	450
Aggiunta di uno stream.....	451
Eliminazione di uno stream.....	452
Aggiunta di un punto di sottoscrizione.....	452
Configurazione delle reti di pubblicazione / sottoscrizione distribuite.....	453
Configurazione di più installazioni.....	472
Connessione di applicazioni in un ambiente di installazione multiplo.....	472
Modifica dell'installazione primaria.....	479
Associazione di un gestore code a un'installazione.....	481
Ricerca di installazioni di IBM MQ su un sistema.....	482
Configurazione dell'alta disponibilità, ripristino e riavvio.....	483
Riconnessione automatica del client.....	484
Console message monitoring.....	491
Configurazioni HA (High Availability).....	495
Registrazione: verifica che i messaggi non vengano persi.....	664
Backup e ripristino dei dati del gestore code IBM MQ.....	696

Modifiche al ripristino da errori cluster sui server su Multiplatforms.....	703
Configurazione di risorse di JMS e Jakarta Messaging.....	705
Configurazione di factory di connessione e destinazioni in un namespace JNDI.....	706
Configurazione di oggetti JMS 2.0 utilizzando IBM MQ Explorer.....	710
Configurazione degli oggetti JMS e Jakarta Messaging utilizzando gli strumenti di amministrazione.....	711
Configurazione di risorse JMS 2.0 in WebSphere Application Server.....	721
Configurazione di WebSphere Application Server per utilizzare il livello di manutenzione dell'adattatore di risorse più recente.....	731
Configurazione della proprietà JMS PROVIDERVERSION	734
Rimozione di sottoscrizioni durevoli WebSphere Application Server.....	741
Configurazione di Managed File Transfer.....	744
Opzioni di configurazione MFT su Multiplatforms.....	744
MFT configuration options on z/OS.....	746
Download e configurazione di Redistributable Managed File Transfer components.....	746
Creating an MFT Agent or Logger command data set.....	752
Configuring Managed File Transfer for z/OS.....	753
Configurazione di MFT su IBM i.....	785
Configurazione di MFT per il primo utilizzo.....	786
Configurazione dei gestori code dell'agent MFT.....	797
Configurazione di un programma di registrazione MFT.....	807
Configurazione del bridge Connect:Direct.....	832
Configurazione di IBM MQ Console e REST API.....	837
Configurazione di base per il server mqweb.....	838
Configurazione del IBM MQ Web Server autonomo.....	842
Configurazione della sicurezza.....	843
Configurazione del nome host HTTP.....	844
Configurazione delle porte HTTPS e HTTP.....	845
Configurazione del timeout di risposta.....	846
Configurazione dell'avvio automatico.....	847
Configurazione della registrazione nei log.....	848
Configurazione del token LTPA.....	851
Configurazione del comportamento della connessione del gestore code remoto per IBM MQ Console.....	853
Configurazione del gateway administrative REST API.....	855
Configurazione di messaging REST API.....	857
Configurazione di REST API per MFT.....	863
Ottimizzazione della JVM del server mqweb.....	868
Struttura del file del componente di installazione IBM MQ Console e REST API.....	870
Backup e ripristino della configurazione del server mqweb.....	872
Definizione di una connessione Aspera gateway su piattaforme Linux o Windows.....	874
Configurazione di IBM MQ per l'utilizzo con il servizio di misurazione IBM Cloud Private.....	878
Configurazione di un gestore code per l'utilizzo con l'istanza del servizio di misurazione su IBM Cloud Private.....	880
Connessione al servizio di misurazione IBM Cloud Private tramite un proxy HTTP.....	882
Risoluzione dei problemi di connessione al servizio di misurazione.....	883
Configuring queue managers on z/OS.....	883
Preparing to customize queue managers on z/OS.....	884
Setting up IBM MQ for z/OS.....	888
Testing a queue manager on z/OS.....	953
Setting up communications with other queue managers on z/OS.....	961
Using IBM MQ with IMS.....	991
Using IBM MQ with CICS.....	999
Upgrading and applying service to Language Environment or z/OS Callable Services.....	999
Using OTMA exits in IMS.....	1001
Using IBM z/OSMF to automate IBM MQ.....	1005
Abilitazione degli agent MFT per connettersi a gestori code z/OS remoti.....	1016
Configurazione di IBM MQ Internet Pass-Thru.....	1017

Supporto per HTTP in MQIPT.....	1017
Supporto SOCKS in MQIPT.....	1019
Supporto SSL/TLS in MQIPT.....	1020
Java security manager in MQIPT.....	1049
Uscite di sicurezza in MQIPT.....	1052
Controllo numero porta in MQIPT.....	1056
Crittografia delle password memorizzate in MQIPT.....	1056
Altre considerazioni sulla sicurezza per MQIPT.....	1058
Log di connessione in MQIPT.....	1059
Configurazione di IBM MQ Internet Pass-Thru utilizzando i container.....	1060
Configurazione delle code di flusso.....	1060

Informazioni particolari..... 1063

Informazioni sull'interfaccia di programmazione.....	1064
Marchi.....	1064

Configurazione di IBM MQ

Creare uno o più gestori code su uno o più computer e configurarli sui sistemi di sviluppo, test e produzione per elaborare i messaggi che contengono i dati di business.

Informazioni su questa attività

Prima di configurare IBM MQ, leggere i concetti IBM MQ in [IBM MQ Panoramica tecnica](#). Leggi come pianificare il tuo ambiente IBM MQ in [Pianificazione](#).

Esistono diversi metodi che è possibile utilizzare per creare, configurare e amministrare i gestori code e le relative risorse in IBM MQ. Questi metodi includono le interfacce della riga comandi, una GUI e un'API di gestione. Per ulteriori informazioni su queste interfacce, consultare [Amministrazione IBM MQ](#).


Per istruzioni su come creare, avviare, arrestare ed eliminare un gestore code, consultare [“Creazione di gestori code su più piattaforme”](#) a pagina 7.

Per informazioni su come creare i componenti richiesti per collegare le applicazioni e le installazioni IBM MQ, consultare [“Configurazione dell'accodamento distribuito”](#) a pagina 205.

Per istruzioni su come collegare i propri client a un server IBM MQ utilizzando metodi diversi, consultare [“Configurazione delle connessioni tra client e server”](#) a pagina 16.

Per istruzioni su come configurare un cluster di gestori code, consultare [“Configurazione di un cluster di gestore code”](#) a pagina 303.

È possibile modificare il comportamento di IBM MQ o di un gestore code modificando le informazioni di configurazione. Per ulteriori informazioni, consultare [“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms”](#) a pagina 93. In generale, non è necessario riavviare un gestore code per rendere effettive le modifiche di configurazione, ad eccezione di quando indicato nella documentazione di questo prodotto.

 Per istruzioni su come configurare IBM MQ for z/OS, consultare [“Configuring queue managers on z/OS”](#) a pagina 883.

Concetti correlati

[IBM MQ - Sommario tecnico](#)

Attività correlate

[Amministrazione di oggetti IBM MQ locali](#)

[Gestione di oggetti IBM MQ remoti](#)

 [Amministrazione di IBMi](#)

 [Amministrazione IBM MQ for z/OS](#)

[Pianificazione](#)

 [Pianificazione dell'ambiente IBM MQ su z/OS](#)

[“Configuring queue managers on z/OS”](#) a pagina 883

Use these instructions to configure queue managers on IBM MQ for z/OS.

Creazione di gestori code su più piattaforme

Prima di poter utilizzare messaggi e code, è necessario creare e avviare almeno un gestore code e i relativi oggetti associati. Un gestore code gestisce le risorse associate ad esso, in particolare le code di sua proprietà. Fornisce servizi di accodamento alle applicazioni per chiamate e comandi MQI (Message queuing Interface) per creare, modificare, visualizzare ed eliminare oggetti IBM MQ.

Prima di iniziare

Importante: IBM MQ non supporta i nomi macchina che contengono spazi. Se si installa IBM MQ su un computer con un nome macchina che contiene spazi, non è possibile creare alcun gestore code.

Prima di poter creare un gestore code, è necessario considerare diversi punti, soprattutto in un ambiente di produzione. Utilizzare il seguente elenco di controllo:

L'installazione associata al gestore code

Per creare un gestore code, utilizzare il IBM MQ comando di controllo `crtmqm`. Il comando `crtmqm` associa automaticamente un gestore code all'installazione da cui è stato immesso il comando `crtmqm`. Per i comandi che operano su un gestore code, è necessario immettere il comando dall'installazione associata al gestore code. È possibile modificare l'installazione associata di un gestore code utilizzando il comando `setmqm`. Tenere presente che il programma di installazione di Windows non aggiunge l'utente che esegue l'installazione al gruppo `mqm`, per ulteriori dettagli, consultare [Autorità per amministrare IBM MQ su AIX, Linux®, and Windows](#).

Convenzioni di denominazione

Utilizzare nomi in MAIUSCOLO per poter comunicare con gestori code di qualsiasi piattaforma. Tenere presente che i nomi vengono assegnati esattamente come vengono immessi. Per evitare l'inconveniente di un sacco di digitazione, non utilizzare nomi inutilmente lunghi.

Specificare un nome gestore code univoco

Quando si crea un gestore code, assicurarsi che nessun altro gestore code abbia lo stesso nome nella rete. I nomi dei gestori code non vengono controllati quando viene creato il gestore code e i nomi che non sono univoci non consentono di creare canali per l'accodamento distribuito. Inoltre, se si utilizza la rete per la messaggistica di pubblicazione / sottoscrizione, le sottoscrizioni vengono associate al nome del gestore code che le ha create. Pertanto, se i gestori code nel cluster o nella gerarchia hanno lo stesso nome, è possibile che le pubblicazioni non le raggiungano.

Un modo per garantire l'univocità consiste nel far precedere ogni nome di gestore code con il proprio nome nodo univoco. Ad esempio, se un nodo è denominato `ACCOUNTS`, è possibile denominare il proprio gestore code `ACCOUNTS.SATURN.QUEUE.MANAGER`, dove `SATURN` identifica un determinato gestore code e `QUEUE.MANAGER` è un'estensione che è possibile assegnare a tutti i gestori code. In alternativa, è possibile omettere questa opzione, ma tenere presente che `ACCOUNTS.SATURN` e `ACCOUNTS.SATURN.QUEUE.MANAGER` sono nomi di gestori code differenti.

Se si sta utilizzando IBM MQ per la comunicazione con altre aziende, è anche possibile includere il proprio nome aziendale come prefisso. Questo non è mostrato negli esempi, perché li rende più difficili da seguire.

Nota: I nomi dei gestori code nei comandi di controllo sono sensibili al maiuscolo / minuscolo. Ciò significa che è consentito creare due gestori code con i nomi `jupiter.queue.manager` e `JUPITER.queue.manager`. Tuttavia, è meglio evitare tali complicazioni.

Limita numero di gestori code

È possibile creare il numero di gestori code consentito dalle risorse. Tuttavia, poiché ciascun gestore code richiede le proprie risorse, è generalmente meglio avere un gestore code con 100 code su un nodo piuttosto che dieci gestori code con dieci code ciascuno.

Nei sistemi di produzione, molti processori possono essere utilizzati con un singolo gestore code, ma le macchine server più grandi potrebbero essere eseguite in modo più efficace con più gestori code.

Specificare un gestore code predefinito

Ogni nodo deve disporre di un gestore code predefinito, anche se è possibile configurare IBM MQ su un nodo senza un gestore code. Il gestore code predefinito è il gestore code a cui si connettono le applicazioni se non specificano un nome di gestore code in una chiamata `MQCONN`. È anche il gestore code che elabora i comandi `MQSC` quando si richiama il comando `runmqsc` senza specificare il nome del gestore code.

Se si specifica un gestore code come predefinito, quest'ultimo sostituisce qualsiasi specifica del gestore code predefinito per il nodo.

La modifica della gestione della coda predefinita può influire su altri utenti o applicazioni. La modifica non ha alcun effetto sulle applicazioni attualmente connesse, perché possono utilizzare l'handle dalla loro chiamata di connessione originale in qualsiasi ulteriore chiamata MQI. Questo handle garantisce che le chiamate siano dirette allo stesso gestore code. Tutte le applicazioni che si collegano *dopo* aver modificato il gestore code predefinito si connettono al nuovo gestore code predefinito. Questo potrebbe essere ciò che si intende, ma è necessario tenerne conto prima di modificare il valore predefinito.

La creazione di un gestore code predefinito è descritta in [“Creazione di un gestore code predefinito”](#) a pagina 12.

Specificare una coda di messaggi non instradabili

La coda di messaggi non instradati è una coda locale in cui i messaggi vengono inseriti se non possono essere instradati alla destinazione desiderata.

È importante disporre di una coda messaggi non recapitabili su ciascun gestore code della rete in uso. Se non se ne definisce una, gli errori che si verificano nei programmi applicativi potrebbero comportare la chiusura dei canali e le risposte ai comandi di gestione potrebbero non essere ricevute.

Ad esempio, se un'applicazione tenta di inserire un messaggio in una coda su un altro gestore code, ma fornisce il nome coda errato, il canale viene arrestato e il messaggio rimane nella coda di trasmissione. Altre applicazioni non possono quindi utilizzare questo canale per i loro messaggi.

I canali non vengono influenzati se i gestori code hanno code di messaggi non recapitabili. Il messaggio non consegnato viene inserito nella coda di messaggi non recapitabili all'estremità ricevente, lasciando il canale e la relativa coda di trasmissione disponibili.

Quando si crea un gestore code, utilizzare l'indicatore **-u** per specificare il nome della coda di messaggi non recapitabili. È anche possibile utilizzare un comando MQSC per modificare gli attributi di un gestore code già definito per specificare la coda di messaggi non instradabili da utilizzare. Fare riferimento a [Visualizzazione e modifica degli attributi del gestore code](#) per un esempio del comando MQSC ALTER.

Specificare una coda di trasmissione predefinita

Una coda di trasmissione è una coda locale in cui i messaggi in transito verso un gestore code remoto sono accodati prima della trasmissione. La coda di trasmissione predefinita è la coda utilizzata quando non viene definita esplicitamente nessuna coda. A ciascun gestore code è possibile assegnare una coda di trasmissione predefinita.

Quando si crea un gestore code, utilizzare l'indicatore **-d** per specificare il nome della coda di trasmissione predefinita. Ciò non crea effettivamente la coda; è necessario farlo esplicitamente in un secondo momento. Per ulteriori informazioni, consultare [Gestione delle code locali](#).

Specificare i parametri di registrazione richiesti

È possibile indicare i parametri di registrazione nel comando `crtmqm`, incluso il tipo di registrazione, il percorso e la dimensione dei file di log.

In un ambiente di sviluppo, i parametri di registrazione predefiniti devono essere adeguati. Tuttavia, è possibile modificare i valori predefiniti se, ad esempio:

- Si dispone di una configurazione di sistema di basso livello che non può supportare log di grandi dimensioni.
- Si prevede che un numero elevato di messaggi lunghi si trovino contemporaneamente nelle code.
- Si prevedono molti messaggi persistenti che passano attraverso il gestore code.

Dopo aver impostato i parametri di registrazione, alcuni di essi possono essere modificati solo eliminando il gestore code e ricreandolo con lo stesso nome ma con parametri di registrazione differenti.

Per ulteriori informazioni sui parametri di registrazione, consultare [“Registrazione: verifica che i messaggi non vengano persi”](#) a pagina 664.

È possibile creare la directory del gestore code `/var/mqm/qmgrs/qmgr`, anche su un filesystem locale separato, prima di utilizzare il comando `crtmqm`. Quando si utilizza `crtmqm`, se la directory `/var/mqm/qmgrs/qmgr` esiste, è vuota ed è di proprietà di `mqm`, viene utilizzata per i dati del gestore code. Se la directory non è di proprietà di `mqm`, la creazione non riesce con un First Failure Support Technology (FFST) messaggio. Se la directory non è vuota, viene creata una nuova directory.

Informazioni su questa attività

Per creare un gestore code, utilizzare il IBM MQ comando di controllo `crtmqm`. Per ulteriori informazioni, consultare `crtmqm`. Il comando `crtmqm` crea automaticamente gli oggetti predefiniti richiesti e gli oggetti di sistema (consultare [Oggetti predefiniti di sistema](#)). Gli oggetti predefiniti formano la base di tutte le definizioni di oggetto che si creano; gli oggetti di sistema sono richiesti per l'operazione del gestore code.

Sui sistemi Windows è possibile avviare più istanze del gestore code utilizzando l'opzione `sax` del comando `crtmqm`.

Una volta creato il Gestore code e i suoi oggetti, è possibile utilizzare il comando `strmqm` per avviare il gestore code.

Procedura

- Per informazioni sulla creazione e la gestione dei gestori code, consultare i topic secondari riportati di seguito:
 - [“Creazione di un gestore code predefinito” a pagina 12](#)
 - [“Impostazione di un gestore code esistente come predefinito” a pagina 13](#)
 - [“Backup dei file di configurazione dopo la creazione di un gestore code” a pagina 14](#)

Concetti correlati

[Uso dei gestori code](#)

Attività correlate

[Creazione di un gestore code denominato QM1](#)

[“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms” a pagina 93](#)

È possibile modificare il comportamento di IBM MQ o di un singolo gestore code per adattarlo alle esigenze della propria installazione modificando le informazioni nei file di configurazione (.ini). È anche possibile modificare le opzioni di configurazione per IBM MQ MQI clients.

[“Configuring queue managers on z/OS” a pagina 883](#)

Use these instructions to configure queue managers on IBM MQ for z/OS.

Riferimenti correlati

[Oggetti di sistema e predefiniti](#)

[crtmqm](#)

Directory effimera configurabile

La directory effimera configurabile definisce l'ubicazione in cui devono andare i dati effimeri per il gestore code. Questo può essere utilizzato per consentire ai socket del dominio AIX and Linux di essere posizionati su un filesystem non montato in un ambiente Red Hat® OpenShift®.

Prima di IBM MQ 9.2.0, sulle piattaforme AIX and Linux, quando un gestore code è in esecuzione, i socket di dominio AIX and Linux vengono creati nella directory `/var/mqm/sockets`. Quando si esegue il gestore code in un contenitore, con `/var/mqm` come file system montato, alcune piattaforme Linux possono impedire la creazione di questi socket di dominio, poiché consentono ad alcuni processi esterni al contenitore di interferire con le operazioni all'interno del contenitore. Questo problema impedisce

l'esecuzione di IBM MQ in una piattaforma del contenitore Red Hat OpenShift , nel contesto di sicurezza predefinito.

Da IBM MQ 9.2.0, l'attributo **EphemeralPrefix** può essere utilizzato per configurare l'ubicazione della directory effimera. Se non si utilizza questo attributo, non verrà visualizzato alcun cambiamento nel comportamento.

Quando una voce del gestore code viene creata in `mqs.ini` (utilizzando i comandi **crtmqm** o **addmqinf**), l'attributo **EphemeralPrefix** viene aggiunto se:

- Impostare l'attributo **DefaultEphemeralPrefix** in “Stanza AllQueueManagers del file mqs.ini” a pagina 99.
- Impostare la variabile di ambiente **MQ_EPHEMERAL_PREFIX**.
- Specificare **-v EphemeralPrefix** solo per il comando **addmqinf**.

È anche possibile aggiungere esplicitamente l'attributo **EphemeralPrefix** a un gestore code esistente quando viene arrestato e questo viene aggiunto quando il gestore code viene riavviato.




Se si specifica un attributo **EphemeralPrefix**, quando il gestore code viene avviato, i dati effimeri per il gestore code vengono creati con tale prefisso anziché con la normale ubicazione. Cioè:

- I file socket generalmente presenti in `/var/mqm/sockets/<QM>` ora si trovano in `/<EphemeralPrefix>/sockets/<QM>`
- I file del pool secondario solitamente presenti in `/<Prefix>/qmgrs/<QM>/@<Subpool>` ora si trovano in `/<EphemeralPrefix>/qmgrs/<QM>/@<Subpool>`

Note:

- `/var/mqm/sockets/@SYSTEM` rimane nella sua posizione fissa e non fa parte dell'attributo **EphemeralPrefix**.
- `AMQCLCHL.TAB` rimane sotto `/<Prefix>/qmgrs/<QM>/@ipcc` e non fa parte dell'attributo **EphemeralPrefix**.

Il numero di caratteri che l'attributo **EphemeralPrefix** può includere dipende dalla propria piattaforma:

-   Sulle piattaforme AIX and Linux è limitato a 12 caratteri.
-  Su IBM i è limitato a 24 caratteri.

Se si specifica un attributo **EphemeralPrefix** che è troppo lungo o non esiste, si riceve un messaggio **AMQ7001E**:

AMQ7001E: L'ubicazione specificata per il gestore code non è valida

.

Multi Directory dati utente

È possibile utilizzare la directory `userdata` per memorizzare lo stato dell'applicazione persistente.

Ogni gestore code IBM MQ dispone di un filesystem dedicato per il relativo stato persistente, che include sia i dati della coda che il log di recupero. Il file system include una directory `userdata` che è possibile utilizzare per memorizzare le informazioni sullo stato persistente per le applicazioni. Consultare [Contenuto della directory su Unix e Linux Systems](#) e [Contenuto della directory su sistemi Windows](#).

La directory `userdata` può essere utile in diverse situazioni, ad esempio:

- Nelle configurazioni RDQM in modo che le informazioni dell'applicazione vengano spostate anche quando un gestore code esegue il failover su un altro nodo (consultare “[Archiviazione dello stato dell'applicazione persistente](#)” a pagina 605).
- Per i gestori code a più istanze, in modo che lo stato dell'applicazione sia ubicato con i relativi dati del gestore code sul filesystem di rete condiviso.
- Più in generale, dove le applicazioni sono servizi del gestore code configurati.

Se si sceglie di memorizzare lo stato dell'applicazione nella directory `userdata`, è necessario tenere presente che i dati scritti in questa ubicazione potrebbero consumare lo spazio su disco disponibile assegnato al gestore code. È necessario assicurarsi che sia disponibile spazio su disco sufficiente per il gestore code per scrivere i dati della coda, i log e altre informazioni sullo stato persistente.

La directory `userdata` ha la proprietà utente e gruppo `mqm` ed è leggibile in modo che gli utenti possano accedervi senza dover far parte del gruppo di amministratori IBM MQ (`mqm`). Non è possibile modificare le autorizzazioni della directory `userdata`, ma è possibile creare il contenuto in essa con qualsiasi proprietà e autorizzazione richiesti.

Multi Creazione di un gestore code predefinito

Il gestore code predefinito è il gestore code a cui si connettono le applicazioni se non specificano un nome gestore code in una chiamata `MQCONN`. È anche il gestore code che elabora i comandi `MQSC` quando si richiama il comando `runmqsc` senza specificare il nome di un gestore code. Per creare un gestore code, utilizzare il IBM MQ comando di controllo `crtmqm`.

Prima di iniziare

Prima di creare un gestore code predefinito, leggere le considerazioni descritte in [“Creazione di gestori code su più piattaforme”](#) a pagina 7.

Linux **AIX** Quando si utilizza `crtmqm` per creare un gestore code su AIX and Linux, se la directory `/var/mqm/qmgrs/qmgr` esiste già, è di proprietà di `mqm` ed è vuota, viene utilizzata per i dati del gestore code. Se la directory non è di proprietà di `mqm`, la creazione del gestore code ha esito negativo con un messaggio First Failure Support Technology (FFST). Se la directory non è vuota, viene creata una nuova directory per i dati del gestore code.

Questa considerazione si applica anche quando la directory `/var/mqm/qmgrs/qmgr` esiste già su un file system locale separato.

Informazioni su questa attività

Quando si crea un gestore code utilizzando il comando `crtmqm`, il comando crea automaticamente gli oggetti predefiniti richiesti e gli oggetti di sistema. Gli oggetti predefiniti formano la base di tutte le definizioni di oggetto che si creano e gli oggetti di sistema sono richiesti per l'operazione del gestore code.

Includendo i parametri rilevanti nel comando, è anche possibile definire, ad esempio, il nome della coda di trasmissione predefinita che deve essere utilizzata dal gestore code e il nome della coda di messaggi non recapitabili.

Windows Su Windows, è possibile utilizzare l'opzione `sax` del comando `crtmqm` per avviare più istanze del gestore code.

Per ulteriori informazioni sul comando `crtmqm` e la sua sintassi, consultare [crtmqm](#).

Procedura

- Per creare un gestore code predefinito, utilizzare il comando `crtmqm` con l'indicatore `-q`.

Il seguente esempio del comando `crtmqm` crea un gestore code predefinito denominato `SATURN.QUEUE.MANAGER`:

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE SATURN.QUEUE.MANAGER
```

dove:

-q

Indica che questo gestore code è il gestore code predefinito.

-d MY.DEFAULT.XMIT.QUEUE

Indica il nome della coda di trasmissione predefinita che deve essere utilizzata da questo gestore code.

Nota: IBM MQ non crea una coda di trasmissione predefinita per l'utente; è necessario definirla personalmente.

-u SYSTEM.DEAD.LETTER.QUEUE

È il nome della coda di messaggi non instradabili predefinita creata da IBM MQ durante l'installazione.

SATURN.QUEUE.MANAGER

È il nome di questo gestore code. Deve essere l'ultimo parametro specificato nel comando `crtmqm`.

Operazioni successive

Una volta creato il gestore code e i suoi oggetti, utilizzare il comando `strmqm` per [Avviare il gestore code](#).

Concetti correlati

[Gestione delle code locali](#)

Attività correlate

[“Backup dei file di configurazione dopo la creazione di un gestore code” a pagina 14](#)

Le informazioni di configurazione IBM MQ sono memorizzate nei file di configurazione su AIX, Linux, and Windows. Dopo aver creato un gestore code, eseguire il backup dei propri file di configurazione. Quindi, se si crea un altro gestore code che causa problemi, è possibile ripristinare i backup una volta rimossa l'origine del problema.

[Visualizzazione e modifica degli attributi del gestore code](#)

Riferimenti correlati

[Oggetti di sistema e predefiniti](#)

Multi **Impostazione di un gestore code esistente come predefinito**

È possibile rendere un gestore code esistente il gestore code predefinito manualmente utilizzando un editor di testo oppure, su Windows e Linux, utilizzando IBM MQ Explorer.

Informazioni su questa attività

Per utilizzare un editor di testo per rendere un gestore code esistente il gestore code predefinito, completare la seguente procedura.

Linux **Windows** Su sistemi Windows e Linux (piattaformex86 e x86-64), se si preferisce utilizzare IBM MQ Explorer per apportare questa modifica, consultare [“Utilizzo di IBM MQ Explorer per impostare un gestore code come predefinito” a pagina 14](#).

Quando si crea un gestore code predefinito, il nome viene inserito nell'attributo Name della stanza `DefaultQueueManager` nel file di configurazione IBM MQ (`mqs.ini`). La stanza e il relativo contenuto vengono creati automaticamente se non esistono.

Procedura

- Per impostare un gestore code esistente come predefinito, modificare il nome del gestore code nell'attributo Name con il nome del nuovo gestore code predefinito. È possibile eseguire questa operazione manualmente, utilizzando un editor di testo.
- Se non si dispone di un gestore code predefinito sul nodo e si desidera rendere predefinito un gestore code esistente, creare la stanza `DefaultQueueManager` con il nome richiesto.

- Se si imposta accidentalmente un altro gestore code come predefinito e si desidera ripristinare il gestore code predefinito originale, modificare la sezione `DefaultQueueManager` in `mqs.ini`, sostituendo il gestore code predefinito indesiderato con quello desiderato.

Attività correlate

“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms” a pagina 93
 È possibile modificare il comportamento di IBM MQ o di un singolo gestore code per adattarlo alle esigenze della propria installazione modificando le informazioni nei file di configurazione (.ini). È anche possibile modificare le opzioni di configurazione per IBM MQ MQI clients.

Linux Windows Utilizzo di IBM MQ Explorer per impostare un gestore code come predefinito

Su sistemi Windows e Linux (x86 e x86-64), è possibile utilizzare IBM MQ Explorer per rendere un gestore code esistente il gestore code predefinito.

Informazioni su questa attività

Per utilizzare IBM MQ Explorer per rendere un gestore code esistente il gestore code predefinito sui sistemi Windows e Linux (piattaformex86 e x86-64), completare la seguente procedura.

Se si preferisce utilizzare un editor di testo per apportare questa modifica manualmente, consultare “Impostazione di un gestore code esistente come predefinito” a pagina 13.

Procedura

1. Aprire IBM MQ Explorer.
2. Fare clic con il tasto destro del mouse su **IBM MQ**, quindi selezionare **Proprietà**. Viene visualizzato il pannello **Proprietà per IBM MQ**.
3. Immettere il nome del gestore code predefinito nel campo **Nome gestore code predefinito**.
4. Fare clic su **OK**.

ALW Backup dei file di configurazione dopo la creazione di un gestore code

Le informazioni di configurazione IBM MQ sono memorizzate nei file di configurazione su AIX, Linux, and Windows. Dopo aver creato un gestore code, eseguire il backup dei propri file di configurazione. Quindi, se si crea un altro gestore code che causa problemi, è possibile ripristinare i backup una volta rimossa l'origine del problema.

Informazioni su questa attività

Come regola generale, eseguire il backup dei file di configurazione ogni volta che si crea un nuovo gestore code.

Esistono due tipi di file di configurazione:

- Quando si installa il prodotto, viene creato il file di configurazione IBM MQ (`mqs.ini`). Contiene un elenco di gestori code che viene aggiornato ogni volta che si crea o si elimina un gestore code. È presente un file `mqs.ini` per nodo.
- Quando si crea un nuovo gestore code, viene creato automaticamente un nuovo file di configurazione del gestore code (`qm.ini`). Contiene i parametri di configurazione per il gestore code.

Se è stato installato il servizio AMQP, è necessario eseguire il backup di un altro file di configurazione:

- **Windows** Su sistemi Windows : `amqp_win.properties`
- **Linux AIX** Su sistemi AIX and Linux : `amqp_unix.properties`

Attività correlate

“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms” a pagina 93

È possibile modificare il comportamento di IBM MQ o di un singolo gestore code per adattarlo alle esigenze della propria installazione modificando le informazioni nei file di configurazione (.ini). È anche possibile modificare le opzioni di configurazione per IBM MQ MQI clients.

“Backup e ripristino dei dati del gestore code IBM MQ” a pagina 696

È possibile proteggere i gestori code da possibili danneggiamenti causati da errori hardware eseguendo il backup dei gestori code e dei dati del gestore code, eseguendo solo il backup della configurazione del gestore code e utilizzando un gestore code di backup.

Spostamento di un gestore code su un sistema operativo differente

Seguire queste istruzioni per spostare un gestore code da un sistema operativo all'altro. Si noti che **non** si tratta di una migrazione di un gestore code.

Informazioni su questa attività

Spostare un gestore code ricreandolo sul sistema di destinazione. La procedura ricrea la configurazione del gestore code, non tenta di ricreare lo stato corrente del gestore code, ad esempio scaricando e ricaricando le code.

Procedura

1. Accedere al sistema di origine come utente nel gruppo di amministratori IBM MQ (mqm).
2. Salvare le informazioni di configurazione del gestore code che si desidera spostare immettendo il seguente comando:

```
dmpmqcfg -a -m QM_name > QM_file
```



dove:

- *QM_name* è il nome del gestore code che si desidera spostare.
- *QM_file* è il nome e il percorso di un file locale sul sistema di origine in cui vengono scritte le informazioni di configurazione.

Per ulteriori informazioni, consultare **dmpmqcfg**.

3. Se il gestore code fa parte di una configurazione distribuita, disattivare il gestore code. Assicurarsi che non vi siano messaggi in corso, quindi [arrestare il gestore code](#).
4. Se si sta passando da una versione del prodotto ad un'altra, migrare il gestore code sul sistema operativo corrente all'ultima versione.

Se il sistema operativo corrente è:

-  Windows, consultare [Migrazione di un gestore code su Windows](#)
-  Linux o AIX, fare riferimento a [Migrazione di un gestore code su AIX and Linux](#)

È necessario assicurarsi che le applicazioni esistenti funzionino ancora.

5. Creare un gestore code vuoto sul sistema operativo utilizzando **crtmqm**.
6. Copiare le definizioni oggetto nel nuovo gestore code appena creato, utilizzando **dmpmqcfg**.

È necessario prestare molta attenzione quando si copiano le definizioni di oggetto, poiché potrebbe essere necessaria una modifica manuale delle definizioni:

- È necessario controllare diversi attributi e potrebbe essere necessario modificarli. Eccone alcune:
 - Indirizzi IP e porte in canali, listener e altri oggetti
 - Informazioni di sicurezza come gli ID utente

- **startcmd** nei servizi
- Vari altri attributi.
- I sottoscrittori durevoli non gestiti potrebbero perdere i messaggi.
- Potrebbe essere necessario modificare anche altri gestori code in modo che i relativi canali si connettano al gestore code spostato.

Una volta copiate le definizioni, i messaggi dell'applicazione dal gestore code, sul sistema operativo originale, devono essere copiati nel gestore code sul nuovo sistema operativo, utilizzando un'applicazione che sposta i messaggi. Quindi verificare che le applicazioni funzionino ancora.

Configurazione delle connessioni tra client e server

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

Informazioni su questa attività

In IBM MQ, i collegamenti di comunicazioni logiche tra oggetti sono denominati *canali*. I canali utilizzati per collegare IBM MQ MQI clients a server sono denominati canali MQI. Si configurano le definizioni di canale a ciascuna estremità del proprio collegamento in modo che l'applicazione IBM MQ sul IBM MQ MQI client possa comunicare con il gestore code sul server.

Prima di definire i canali MQI, è necessario decidere quale forma di comunicazione utilizzare e definire la connessione ad ogni estremità del canale.

Se si sta definendo un canale MQI tra un IBM MQ MQI client e un gestore code che si trovano su reti fisiche differenti o che comunicano tramite firewall, l'utilizzo di IBM MQ Internet Pass-Thru potrebbe semplificare la configurazione. Per ulteriori informazioni, consultare [IBM MQ Internet Pass-Thru](#).

Procedura

1. Decidi il tipo di comunicazione che stai per utilizzare.
Consultare [“Il tipo di comunicazione da utilizzare”](#) a pagina 17.
2. Definire la connessione ad ogni estremità del canale.
Per definire la connessione, è necessario:
 - a) Configurare la connessione.
 - b) Registrare i valori dei parametri necessari per le definizioni di canali.
 - c) Abilitare il server per rilevare le richieste di rete in entrata da IBM MQ MQI client, avviando un *listener*.

Concetti correlati

[“IBM MQ MQI client file di configurazione, mqclient.ini”](#) a pagina 168

È possibile configurare i client utilizzando gli attributi in un file di testo. Questi attributi possono essere sovrascritti dalle variabili di ambiente o in altri modi specifici della piattaforma.

Attività correlate

[“Utilizzo delle variabili d'ambiente IBM MQ”](#) a pagina 64

È possibile utilizzare i comandi per visualizzare le impostazioni correnti o per reimpostare i valori delle variabili di ambiente IBM MQ .

[Connessione delle applicazioni client MQI IBM MQ ai gestori code](#)

Riferimenti correlati

[VISUALIZZA CHLAUTH](#)

[SET CHLAUTH](#)

Il tipo di comunicazione da utilizzare

Piattaforme differenti supportano protocolli di comunicazione differenti. La scelta del protocollo di trasmissione dipende dalla combinazione di IBM MQ MQI client e delle piattaforme server.






















Tipi di protocollo di trasmissione per i canali MQI

A seconda delle piattaforme client e server, ci sono fino a quattro tipi di protocollo di trasmissione per i canali MQI:



- TCP/IP
- LU 6.2
- NetBIOS
- SPX

Quando si definiscono i canali MQI, ciascuna definizione di canale deve specificare un attributo del protocollo di trasmissione (tipo di trasporto). Un server non è limitato a un solo protocollo, quindi diverse definizioni di canale possono specificare protocolli differenti. Per IBM MQ MQI clients, può essere utile disporre di canali MQI alternativi che utilizzano protocolli di trasmissione differenti.

La scelta del protocollo di trasmissione dipende anche dalla particolare combinazione di piattaforme client e server IBM MQ . Le combinazioni possibili sono mostrate nella seguente tabella.

Tabella 1. Protocolli di trasmissione - combinazione di piattaforme server e IBM MQ MQI client		
Protocollo di trasmissione	IBM MQ MQI client	IBM MQ server
TCP/IP "1" a pagina 17	 IBM i  AIX  Linux  Windows	 IBM i  AIX  Linux  Windows  z/OS
LU 6.2	 AIX  Linux "2" a pagina 18  Windows	 IBM i  AIX  Linux "2" a pagina 18  Windows  z/OS
NetBIOS	 Windows	 Windows
SPX	 Windows	 Windows

Note:

1.   Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la

velocità di trasmissione di rete. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

2. Tranne Linux (piattaforma POWER)

Concetti correlati

[“Definizione di un collegamento TCP su Windows” a pagina 269](#)

Definire una connessione TCP configurando un canale all'estremità di invio per specificare l'indirizzo della destinazione ed eseguendo un programma listener all'estremità di ricezione.

[“Definizione di un collegamento TCP su AIX and Linux” a pagina 277](#)

La definizione del canale all'estremità di invio specifica l'indirizzo della destinazione. Il listener o il daemon inet è configurato per la connessione all'estremità di ricezione.

[“Definizione di un collegamento TCP su IBM i” a pagina 296](#)

È possibile definire una connessione TCP all'interno della definizione di canale utilizzando il campo Nome connessione.

[“Defining a TCP connection on z/OS” a pagina 982](#)

To define a TCP connection, there are a number of settings to configure.

[“Definizione di una connessione LU 6.2 su Windows” a pagina 271](#)

SNA deve essere configurato in modo che sia possibile stabilire una conversazione LU 6.2 tra le macchine.

[“Definizione di una connessione LU 6.2 su AIX and Linux” a pagina 280](#)

SNA deve essere configurato in modo che sia possibile stabilire una conversazione LU 6.2 tra le macchine.

[“Definizione di una connessione LU 6.2 su IBM i” a pagina 298](#)

Definire i dettagli delle comunicazioni LU 6.2 utilizzando un nome modalità, un nome TP e un nome connessione di una connessione LU 6.2 completa.

[“Definizione di una connessione NetBIOS su Windows” a pagina 272](#)

Una connessione NetBIOS si applica solo a un client e a un server su cui è in esecuzione Windows. IBM MQ utilizza tre tipi di risorsa NetBIOS quando stabilisce una connessione NetBIOS a un altro prodotto IBM MQ : sessioni, comandi e nomi. Ciascuna di queste risorse ha un limite, che viene stabilito per impostazione predefinita o per scelta durante l'installazione di NetBIOS.

Attività correlate

[“Definizione di una connessione Aspera gateway su piattaforme Linux o Windows” a pagina 874](#)

IBM Aspera faspio Gateway fornisce un tunnel TCP/IP veloce che può aumentare in modo significativo la velocità di trasmissione di rete per IBM MQ. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linux o Windows.

Riferimenti correlati

[“Limiti di connessione TCP/IP” a pagina 18](#)

Il numero di richieste di connessione in sospeso che possono essere accodate su una singola porta TCP/IP dipende dalla piattaforma. Si verifica un errore se viene raggiunto il limite.

[“Defining an LU6.2 connection for z/OS using APPC/MVS” a pagina 985](#)







To define an LU6.2 connection there are a number of settings to configure.

Limiti di connessione TCP/IP

Il numero di richieste di connessione in sospeso che possono essere accodate su una singola porta TCP/IP dipende dalla piattaforma. Si verifica un errore se viene raggiunto il limite.

Questo limite di connessione non corrisponde al numero massimo di client che è possibile collegare a un server IBM MQ . È possibile collegare più client a un server, fino al livello determinato dalle risorse di sistema del server. I valori di backlog per le richieste di connessione sono riportati nella seguente tabella:

Tabella 2. Numero massimo di richieste di connessione in sospeso accodate su una porta TCP/IP

Piattaforma server	Numero massimo di richieste di connessione
 AIX	100
 Linux	100
 IBM i	255
Server  Windows	100
Workstation  Windows	100
 z/OS	255

Se viene raggiunto il limite di connessione, il client riceve un codice di ritorno MQRC_HOST_NOT_AVAILABLE dalla chiamata MQCONN e un errore AMQ9202 nel log degli errori del client (/var/mqm/errors/AMQERR0n.LOG su sistemi AIX and Linux o amqerr0n.log nella sottodirectory degli errori dell'installazione del client IBM MQ su Windows). Se il client ritenta la richiesta MQCONN, potrebbe avere esito positivo.

Per aumentare il numero di richieste di connessione che è possibile effettuare ed evitare che i messaggi di errore vengano generati da questa limitazione, è possibile che più listener siano in ascolto su una porta diversa o che abbiano più di un gestore code.

Come configurare un IBM MQ MQI client

Seguire queste istruzioni per configurare un cliente.

Prima di iniziare

Per impostare un IBM MQ MQI client, è necessario disporre di un server IBM MQ già installato e funzionante, a cui si conetterà il client.

Procedura

- Verificare di disporre di una piattaforma adatta per un client IBM MQ MQI e che l'hardware e il software soddisfino i requisiti.
Il supporto piattaforma è descritto in [Supporto piattaforma per client IBM MQ](#).
- Decidere come installare IBM MQ sulla workstation client, quindi seguire le istruzioni per la particolare combinazione di piattaforme client e server.
L'installazione è descritta nei seguenti argomenti:
 -  [Installazione di un client IBM MQ su AIX](#)
 -  [Installazione di un client di IBM MQ su Linux](#)
 -  [Installazione di un client di IBM MQ su Windows](#)
 -  [Installazione di un client di IBM MQ su IBM i](#)
- Verificare che i collegamenti di comunicazione siano configurati e connessi.
La configurazione dei collegamenti di comunicazione è descritta in [Configurazione delle connessioni tra il server e client](#).
- Verificare che l'installazione funzioni correttamente.

Consultare la sezione di verifica della procedura di installazione per la piattaforma o le piattaforme utilizzate dall'azienda.

5. Una volta verificata l'installazione di IBM MQ MQI client , considerare se è necessario proteggere il proprio client.

La sicurezza client è descritta in [Impostazione della sicurezza IBM MQ MQI client](#).

6. Impostare i canali tra il client e il server IBM MQ MQI richiesti dalle applicazioni IBM MQ che si desidera eseguire sul client.

L'impostazione dei canali è descritta in [Definizione di canali MQI](#). Ci sono alcune considerazioni aggiuntive se stai utilizzando TLS.

Queste considerazioni sono descritte in [Specifica che un canale MQI utilizza TLS](#). Potrebbe essere necessario utilizzare un file di configurazione IBM MQ MQI client o variabili di ambiente IBM MQ per impostare i canali. Le variabili di ambiente IBM MQ sono descritte in [Utilizzo delle variabili di ambiente IBM MQ](#).

7. Consultare [Sviluppo delle applicazioni](#) per una descrizione completa delle applicazioni IBM MQ .

8. Quando si progettano, si creano e si eseguono applicazioni nell'ambiente IBM MQ MQI client , è necessario considerare le differenze da un ambiente del gestore code.

Per informazioni su queste differenze, consultare:

- [Utilizzo dell'interfaccia MQI \(Message Queue Interface\) in una applicazione client](#)
- [Creazione di applicazioni per IBM MQ MQI clients](#)
- [Connessione di IBM MQ MQI client applicazioni ai gestori code](#)
- [Risoluzione dei problemi con IBM MQ MQI clients](#)

Configurazione di un client transazionale esteso

Questa raccolta di argomenti descrive come configurare la funzione transazionale estesa per ogni categoria di gestore transazioni.

Per ciascuna piattaforma, il client transazionale esteso fornisce supporto per i seguenti gestori transazioni esterni:

Gestori transazioni compatibili con XA

Il client transazionale esteso fornisce l'interfaccia del gestore risorse XA per supportare i gestori transazioni conformi a XA come CICS e Tuxedo.

Microsoft Transaction Server (solo sistemi Windows)

Solo su sistemi Windows , l'interfaccia del gestore risorse XA supporta anche Microsoft Transaction Server (MTS). Il supporto IBM MQ MTS fornito con il client transazionale esteso fornisce il bridge tra MTS e l'interfaccia del gestore risorse XA.

WebSphere Application Server



WebSphere Application Server 6 e versioni successive include un provider di messaggistica IBM MQ , quindi non è necessario utilizzare il client transazionale esteso.



Configurazione di gestori transazioni compatibili con XA

Per prima cosa, configurare il client di base IBM MQ , quindi configurare la funzione transazionale estesa utilizzando le informazioni in questi argomenti.

Nota: Questa sezione presuppone che l'utente abbia una conoscenza di base dell'interfaccia XA come pubblicata da Open Group in *Distributed Transaction Processing: The XA Specification*.

Per configurare un client transazionale esteso, è necessario prima configurare il client di base IBM MQ come descritto in:

-  [Installazione di un client di IBM MQ su AIX](#)
-  [Installazione di un client di IBM MQ su Linux](#)

-  [Installazione di un client di IBM MQ su Windows](#)
-  [Installazione di un client di IBM MQ su IBM i](#)

Utilizzando le informazioni per la propria piattaforma, è possibile configurare la funzione transazionale estesa per un gestore transazioni compatibile con XA come CICS e Tuxedo.

Un gestore transazioni comunica con un gestore code come gestore risorse utilizzando lo stesso canale MQI utilizzato dall'applicazione client connessa al gestore code. Quando il gestore transazioni emette una chiamata di funzione del gestore risorse (*xa_*), il canale di MQI viene utilizzato per inoltrare la chiamata al gestore code e per ricevere l'output dal gestore code.

Il gestore transazioni può avviare il canale MQI emettendo una chiamata *xa_open* per aprire il gestore code come gestore risorse oppure l'applicazione client può avviare il canale MQI emettendo una chiamata MQCONN o MQCONNX.

- Se il gestore transazioni avvia il canale MQI e l'applicazione client successivamente richiama MQCONN o MQCONNX sullo stesso thread, la chiamata MQCONN o MQCONNX viene completata correttamente e viene restituito un handle di connessione all'applicazione. L'applicazione non riceve un codice di completamento MQCC_WARNING con un codice motivo MQRC_ALREADY_CONNECTED.
- Se l'applicazione client avvia il canale MQI e il gestore transazioni in seguito richiama *xa_open* sullo stesso thread, la chiamata *xa_open* viene inoltrata al gestore code utilizzando tale canale.

In una situazione di ripristino in seguito a un errore, quando non è in esecuzione alcuna applicazione client, il gestore transazioni può utilizzare un canale MQI dedicato per ripristinare le unità di lavoro incomplete a cui il gestore code partecipava al momento dell'errore.

Tenere presenti le seguenti condizioni quando si utilizza un client transazionale esteso con un gestore transazioni compatibile con XA:

- All'interno di un singolo thread, un'applicazione client può essere connessa a un solo gestore code alla volta. Questa limitazione si applica solo quando si utilizza un client transazionale esteso; un'applicazione client che utilizza un client di base IBM MQ può essere connessa a più di un gestore code contemporaneamente all'interno di un singolo thread.
- Ogni thread di una applicazione client può connettersi a un gestore code differente.
- Un'applicazione client non può utilizzare handle di connessione condivisi.

Per configurare la funzione transazionale estesa, è necessario fornire le seguenti informazioni al gestore transazioni per ciascun gestore code che agisce come gestore risorse:

- Una stringa *xa_open*
- Un puntatore a una struttura di commutazione XA

Quando il gestore transazioni richiama *xa_open* per aprire il gestore code come gestore risorse, trasmette la stringa *xa_open* al client transazionale esteso come argomento, *xa_info*, sulla chiamata. Il client transazionale esteso utilizza le informazioni nella stringa *xa_open* nei modi seguenti:

- Per avviare un canale MQI per il gestore code del server, se l'applicazione client non ne ha già avviato uno
- Per verificare che il gestore code che il gestore transazioni apre come gestore risorse sia uguale al gestore code a cui si connette l'applicazione client
- Per individuare le funzioni *ax_reg* e *ax_unreg* del gestore transazioni, se il gestore code utilizza la registrazione dinamica

Per il formato di una stringa *xa_open* e per ulteriori dettagli sul modo in cui le informazioni nella stringa *xa_open* vengono utilizzate da un client transazionale esteso, consultare [“Il formato di una stringa xa_open”](#) a pagina 23.

Una struttura di switch XA consente al gestore transazioni di individuare le funzioni *xa_* fornite dal client transazionale esteso e specifica se il gestore code utilizza la registrazione dinamica. Per informazioni sulle strutture di switch XA fornite con un client transazionale esteso, consultare [“Le strutture di switch XA”](#) a pagina 26.

Per informazioni su come configurare la funzione transazionale estesa per un determinato gestore transazioni e per qualsiasi altra informazione sull'utilizzo del gestore transazioni con un client transazionale esteso, consultare le seguenti sezioni:

- [“Configurazione di un client transazionale esteso per CICS” a pagina 28](#)
- [“Configurazione di un client transazionale esteso per Tuxedo” a pagina 29](#)

Concetti correlati

[“I parametri CHANNEL, TRPTYPE, CONNAME e QMNAME della stringa xa_open” a pagina 25](#)

Utilizzare queste informazioni per comprendere come il client transazionale esteso utilizza questi parametri per determinare il gestore code a cui connettersi.

[“Ulteriore elaborazione degli errori per xa_open” a pagina 26](#)

La chiamata xa_open ha esito negativo in determinate circostanze.

Attività correlate

[“Utilizzo del client transazionale esteso con canali TLS” a pagina 27](#)

Non è possibile impostare un canale TLS utilizzando la stringa xa_open. Seguire queste istruzioni per utilizzare la tabella di definizione del canale client (ccdt).

Riferimenti correlati

[“I parametri TPM e AXLIB” a pagina 25](#)

Un client transazionale esteso utilizza i parametri TPM e AXLIB per individuare le funzioni ax_reg e ax_unreg del gestore transazioni. Queste funzioni vengono utilizzate solo se il gestore code utilizza la registrazione dinamica.

[“Ripristino in seguito a un errore nell'elaborazione transazionale estesa” a pagina 26](#)

In seguito a un errore, un gestore transazioni deve essere in grado di recuperare eventuali unità di lavoro incomplete. A tale scopo, il gestore transazioni deve essere in grado di aprire come gestore risorse qualsiasi gestore code che stava partecipando a un'unità di lavoro incompleta al momento dell'errore.

IBM MQ for z/OS considerations for extended transactional client connections

Some XA transaction managers use sequences of transaction coordination calls which are incompatible with the features normally available to clients connecting to IBM MQ for z/OS.

Where an incompatible sequence is detected, IBM MQ for z/OS might issue an abend for the connection and return an error response to the client.

For example, xa_prepare receives abend 5C6-00D4007D, with return code -3 (XAER_RMERR) returned to the client.

Another example is that xa_end receives abend 5C6-00D40079.

For transaction managers which encounter this situation, take the following action to allow the transaction manager to interact with IBM MQ for z/OS.

Ensure that you have enabled changes to XA client connections on IBM MQ for z/OS which allow the transaction manager to prepare a transaction on a different connection.

Notes:

- The change is not enabled by default. To make use of the change you must specify the keyword CSQSERVICE1 (in upper case) anywhere in the description field of the SVRCONN channel used by the XA client.
- Channels with the CSQSERVICE1 keyword have the following restrictions:
 - GROUP unit of recovery disposition is not permitted. Only QMGR unit of recovery disposition is allowed. The disposition is determined by the name given on the xa_open call. If the queue sharing group name is used, then the XA connection requests a group unit of recovery.

An xa_open call specifying the queue sharing group name in the **xa_info** parameter fails with *xaer_inval*.

- The *MQGMO_LOCK* and *MQGMO_UNLOCK* options are not permitted. An MQGET call with *MQGMO_LOCK* or *MQGMO_UNLOCK* fails with MQRC_ENVIRONMENT_ERROR.

The change was enabled at IBM MQ for z/OS 9.0 through [APAR P173410](#)

Related concepts

“Configurazione di gestori transazioni compatibili con XA” on page 20

Per prima cosa, configurare il client di base IBM MQ , quindi configurare la funzione transazionale estesa utilizzando le informazioni in questi argomenti.

Il formato di una stringa xa_open

Una stringa xa_open contiene coppie di valori e nomi di parametri definiti.

Una stringa xa_open ha il formato seguente:

```
parm_name1 = parm_value1, parm_name2 = parm_value2, ...
```

dove *parm_name* è il nome di un parametro e *parm_value* è il valore di un parametro. I nomi dei parametri non sono sensibili al maiuscolo / minuscolo, ma, se non diversamente specificato, i valori dei parametri sono sensibili al maiuscolo / minuscolo. È possibile specificare i parametri in qualsiasi ordine.

I nomi, i significati e i valori validi dei parametri sono i seguenti:

Nome

Significato e valori validi

CHANNEL

Il nome di un canale MQI.

È un parametro facoltativo. Se viene fornito questo parametro, è necessario fornire anche il parametro CONNAME.

TRPTYPE

Il protocollo di comunicazioni per il canale MQI. I seguenti protocolli sono valori validi:

LU62

SNA LU 6.2

NETBIOS

NetBIOS

SPX

IPX/SPX

TCP

TCP/IP

È un parametro facoltativo. Se viene omesso, viene utilizzato il valore predefinito di TCP. I valori del parametro non sono sensibili al maiuscolo / minuscolo.

CONNAME

L'indirizzo di rete del gestore code all'estremità del server del canale MQI. I valori validi di questo parametro dipendono dal valore del valore del parametro TRPTYPE:

LU62

Un nome di destinazione simbolico, che identifica una voce di informazioni lato CPI-C.

Il nome qualificato di rete di una LU partner non è un valore valido né un alias LU partner. Questo perché non esistono ulteriori parametri per specificare un nome TP (transaction program) e un nome modo.

NETBIOS

Un nome NetBIOS .

SPX

Un indirizzo di rete a 4 byte, un indirizzo di nodo a 6 byte e un numero socket a 2 byte facoltativo. Questi valori devono essere specificati in notazione esadecimale. Un punto deve separare gli

indirizzi di rete e di nodo e il numero di socket, se fornito, deve essere racchiuso tra parentesi. Ad esempio:

```
0a0b0c0d.804abcde23a1(5e86)
```

Se il numero socket viene omissso, viene utilizzato il valore predefinito 5e86 .

TCP

Un nome host o un indirizzo IP, facoltativamente seguito da un numero di porta tra parentesi. Se il numero di porta viene omissso, viene utilizzato il valore predefinito 1414. È possibile specificare più host e porte per un gestore code utilizzando un separatore punto e virgola, ad esempio:

```
host1(1415);host2(1416);host3(1417)
```


È un parametro facoltativo. Se questo parametro viene fornito, deve essere fornito anche il parametro CHANNEL.

QMNAME

Il nome del gestore code all'estremità server del canale MQI. Il nome non può essere vuoto o un singolo asterisco (*), né può iniziare con un asterisco. Ciò significa che il parametro deve identificare un gestore code specifico in base al nome.

Questo è un parametro obbligatorio.

Quando un'applicazione client è connessa a un gestore code specifico, qualsiasi ripristino della transazione deve essere elaborato dallo stesso gestore code.

 Se l'applicazione si connette a un gestore code z/OS , può specificare il nome di un gestore code specifico o il nome di un gruppo di condivisione code (QSG). Utilizzando il nome del gestore code o il nome del gruppo di condivisione code, l'applicazione controlla se partecipa a una transazione con un'unità QMGR di disposizione del ripristino o con un'unità GROUP di disposizione del ripristino. La disposizione dell'unità di recupero GROUP consente il recupero della transazione da elaborare su qualsiasi membro del QSG. Per utilizzare le unità di ripristino GROUP, è necessario abilitare l'attributo gestore code **GROUPUR** . Per ulteriori informazioni sull'utilizzo dell'unità di ripristino GROUP, consultare [Disposizione dell'unità di ripristino in un gruppo di condivisione code](#).

TPM

Il gestore transazioni utilizzato. I valori validi sono CICS e TUXEDO.

Un client transazionale esteso utilizza questo parametro e il parametro AXLIB per lo stesso scopo. Per ulteriori informazioni su questi parametri, consultare [Parametri TPM e AXLIB](#).

È un parametro facoltativo. I valori del parametro non sono sensibili al maiuscolo / minuscolo.

AXLIB

Il nome della libreria che contiene le funzioni ax_reg e ax_unreg del gestore transazioni.

È un parametro facoltativo.

ID utente

L'ID utente fornito per l'autenticazione al gestore code. Se questo parametro viene fornito, è necessario fornire anche il parametro **PWD** . Se l'ID utente e la password forniti sono autenticati, l'ID utente viene utilizzato per identificare la connessione del gestore transazioni. L'ID utente e la password popolano l'oggetto MQCSP sulla chiamata MQCONN.

I parametri **UID** e **PWD** sono validi per i collegamenti client e server.

PWD

La password fornita al gestore code per l'autenticazione. Se questo parametro viene fornito, è necessario fornire anche il parametro **UID** .

Avviso: In alcuni casi, la password in una struttura MQCSP per un'applicazione client verrà inviata attraverso una rete in testo semplice. Per assicurarsi che le password dell'applicazione client siano protette in modo appropriato, consultare [IBM MQCSP password protection](#).

Ecco un esempio di una stringa xa_open:

```
channel=MARS.SVR, trptype=tcp, conname=MARS(1415), qmname=MARS, tpm=cics
```

I parametri CHANNEL, TRPTYPE, CONNAME e QMNAME della stringa xa_open

Utilizzare queste informazioni per comprendere come il client transazionale esteso utilizza questi parametri per determinare il gestore code a cui connettersi.

Se i parametri **CHANNEL** e **CONNAME** vengono forniti nella stringa xa_open, il client transazionale esteso utilizza questi parametri e il parametro **TRPTYPE** per avviare un canale MQI per il gestore code del server.

Se i parametri **CHANNEL** e **CONNAME** non vengono forniti nella stringa xa_open, il client transazionale esteso utilizza il valore della variabile di ambiente MQSERVER per avviare un canale MQI. Se la variabile di ambiente MQSERVER non è definita, il client transazionale esteso utilizza la voce nella definizione di canale client identificata dal parametro **QMNAME**.

In ciascuno di questi casi, il client transazionale esteso verifica che il valore del parametro **QMNAME** sia il nome del gestore code all'estremità del server del canale MQI. Se non lo è, la chiamata xa_open ha esito negativo e il gestore transazioni riporta l'errore all'applicazione.

z/OS Se l'applicazione utilizza un nome del gruppo di condivisione code nel campo del parametro **QMNAME** e la proprietà GROUPUR è disabilitata sul gestore code a cui si connette, la chiamata xa_open non riesce.

z/OS Se il client delle applicazioni si connette a un gestore code z/OS, può specificare un nome QSG (queue sharing group) per il parametro **QMNAME**. Ciò consente al client applicativo di partecipare a una transazione con un'unità di disposizione di recupero GROUP. Per ulteriori informazioni sulla disposizione dell'unità di ripristino GROUP, vedere [Disposizione dell'unità di ripristino](#).

Quando l'applicazione client successivamente richiama MQCONN o MQCONNX sullo stesso thread utilizzato dal gestore transazioni per emettere la chiamata xa_open, l'applicazione riceve un handle di connessione per il canale MQI avviato dalla chiamata xa_open. Un secondo canale MQI non viene avviato. Il client transazionale esteso verifica che il valore del parametro **QMgrName** nella chiamata MQCONN o MQCONNX sia il nome del gestore code all'estremità server del canale MQI. In caso contrario, la chiamata MQCONN o MQCONNX ha esito negativo con un codice motivo MQRC_ANOTHER_Q_MGR_CONNECTED. Se il valore del parametro **QMgrName** è vuoto o un singolo asterisco (*), o inizia con un asterisco, la chiamata MQCONN o MQCONNX ha esito negativo con un codice motivo di MQRC_Q_MGR_NAME_ERROR.

Se l'applicazione client ha già avviato un canale MQI richiamando MQCONN o MQCONNX prima che il gestore transazioni chiami xa_open sullo stesso thread, il gestore transazioni utilizza invece questo canale MQI. Un secondo canale MQI non viene avviato. Il client transazionale esteso verifica che il valore del parametro **QMNAME** nella stringa xa_open sia il nome del gestore code del server. In caso contrario, la chiamata xa_open ha esito negativo.





Se un'applicazione client avvia prima un canale MQI, il valore del parametro **QMgrName** nella chiamata MQCONN o MQCONNX può essere vuoto o un singolo asterisco (*) oppure può iniziare con un asterisco. In queste circostanze, tuttavia, è necessario assicurarsi che il gestore code a cui si connette l'applicazione sia lo stesso gestore code che il gestore transazioni intende aprire come gestore risorse quando in seguito richiama xa_open sullo stesso thread. È possibile che si verifichino meno problemi, quindi, se il valore del parametro **QMgrName** identifica esplicitamente il gestore code per nome.

I parametri TPM e AXLIB

Un client transazionale esteso utilizza i parametri TPM e AXLIB per individuare le funzioni ax_reg e ax_unreg del gestore transazioni. Queste funzioni vengono utilizzate solo se il gestore code utilizza la registrazione dinamica.

Se il parametro TPM viene fornito in una stringa xa_open, ma il parametro AXLIB non viene fornito, il client transazionale esteso assume un valore per il parametro AXLIB basato sul valore del parametro TPM. Consultare [Tabella 3 a pagina 26](#) per i valori assunti del parametro AXLIB.

Tabella 3. Valori assunti del parametro AXLIB

Valore di TPM	Piattaforma	Valore assunto di AXLIB
CICS	 AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	Sistemi  Windows	libEnc
Tuxedo	 AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	Sistemi  Windows	libtux

Se il parametro AXLIB viene fornito in una stringa xa_open, il client transazionale esteso utilizza il suo valore per sovrascrivere qualsiasi valore assunto basato sul valore del parametro TPM. Il parametro AXLIB può essere utilizzato anche per un gestore transazioni per il quale il parametro TPM non ha un valore specificato.

Ulteriore elaborazione degli errori per xa_open

La chiamata xa_open ha esito negativo in determinate circostanze.

Gli argomenti in questa sezione descrivono situazioni in cui la chiamata xa_open ha esito negativo. Ha esito negativo anche se si verifica una delle seguenti situazioni:

- Sono presenti errori nella stringa xa_open.
- Informazioni insufficienti per avviare un canale MQI.
- Si è verificato un problema durante il tentativo di avviare un canale MQI (ad esempio, il gestore code server non è in esecuzione).

Ripristino in seguito a un errore nell'elaborazione transazionale estesa

In seguito a un errore, un gestore transazioni deve essere in grado di recuperare eventuali unità di lavoro incomplete. A tale scopo, il gestore transazioni deve essere in grado di aprire come gestore risorse qualsiasi gestore code che stava partecipando a un'unità di lavoro incompleta al momento dell'errore.

Pertanto, è necessario assicurarsi che tutte le unità di lavoro incomplete siano state risolte prima di apportare modifiche alle informazioni di configurazione.

In alternativa, è necessario verificare che le modifiche di configurazione non influiscano sulla capacità del gestore transazioni di aprire i gestori code che devono essere aperti. Di seguito sono riportati esempi di tali modifiche di configurazione:

- Modifica del contenuto di una stringa xa_open
- Modifica del valore della variabile di ambiente MQSERVER
- Modifica delle voci nella tabella di definizione del canale client (CCDT)
- Eliminazione di una definizione di canale di connessione server

Le strutture di switch XA

Due strutture switch XA vengono fornite con il client transazionale esteso su ciascuna piattaforma.

Queste strutture di switch sono:




MQRMIXASwitch

Questa struttura switch viene utilizzata da un gestore transazioni quando un gestore code, che funge da gestore risorse, non utilizza la registrazione dinamica.

MQRMIXASwitchDynamic

Questa struttura di switch viene utilizzata da un gestore transazioni quando un gestore code, che funge da gestore risorse, utilizza la registrazione dinamica.

Queste strutture di switch si trovano nelle librerie mostrate in [Tabella 4 a pagina 27](#).

Piattaforma	Libreria contenente le strutture di switch XA
 AIX  Linux	MQ_INSTALLATION_PATH/lib/libmqcxa
Sistemi  Windows	MQ_INSTALLATION_PATH\bin\mqcxa.dll ¹

MQ_INSTALLATION_PATH rappresenta la directory di livello superiore in cui è installato IBM MQ .

Il nome del gestore risorse IBM MQ in ogni struttura switch è MQSeries_XA_RMI, ma molti gestori code possono condividere la stessa struttura switch.

Concetti correlati

[“Registrazione dinamica ed elaborazione transazionale estesa” a pagina 27](#)

L'utilizzo della registrazione dinamica è una forma di ottimizzazione perché può ridurre il numero di chiamate alla funzione xa _ emesse dal gestore transazioni.

Registrazione dinamica ed elaborazione transazionale estesa

L'utilizzo della registrazione dinamica è una forma di ottimizzazione perché può ridurre il numero di chiamate alla funzione xa _ emesse dal gestore transazioni.

Se un gestore code non utilizza la registrazione dinamica, un gestore transazioni coinvolge il gestore code in ogni unità di lavoro. Il gestore transazioni esegue questa operazione richiamando xa_start, xa_end e xa_prepare, anche se il gestore code non dispone di risorse aggiornate all'interno dell'unità di lavoro.

Se un gestore code utilizza la registrazione dinamica, un gestore transazioni viene avviato supponendo che il gestore code non sia coinvolto in un'unità di lavoro e non richiami xa_start. Il gestore code viene quindi coinvolto nell'unità di lavoro solo se le relative risorse vengono aggiornate all'interno del controllo del punto di sincronizzazione. Se ciò si verifica, il client transazionale esteso richiama ax_reg per registrare il coinvolgimento del gestore code.

Utilizzo del client transazionale esteso con canali TLS

Non è possibile impostare un canale TLS utilizzando la stringa xa_open. Seguire queste istruzioni per utilizzare la tabella di definizione del canale client (ccdt).

Informazioni su questa attività

A causa della dimensione limitata della stringa xa_open xa_info, non è possibile passare tutte le informazioni richieste per configurare un canale TLS utilizzando il metodo xa_open string di connessione a un gestore code. Pertanto, è necessario utilizzare la tabella di definizione del canale client oppure, se il gestore transazioni lo consente, creare il canale con MQCONNX prima di emettere la chiamata xa_open.

Per utilizzare la tabella di definizione del canale client, seguire queste istruzioni:

Procedura

1. Specificare una stringa xa_open contenente solo il parametro qmname (nome gestore code) obbligatorio, ad esempio: XA_Open_String=qmname=MYQM

2. Utilizzare un gestore code per definire un canale CLNTCONN (client-connection) con i parametri TLS richiesti. Includere il nome gestore code nell'attributo QMNAME nella definizione CLNTCONN. Questo verrà associato al qmname nella stringa xa_open.
3. Rendere la definizione CLNTCONN disponibile per il sistema client in una tabella di definizione del canale client (CCDT) o, su Windows, nella directory attiva.
4. Se si sta utilizzando una CCDT, identificare la CCDT contenente la definizione del canale CLNTCONN utilizzando le variabili di ambiente MQCHLLIB e MQCHLTAB. Impostare queste variabili negli ambienti utilizzati dall'applicazione client e dal gestore transazioni.

Risultati

Ciò fornisce al gestore transazioni una definizione di canale per il gestore code appropriato con gli attributi TLS necessari per eseguire correttamente l'autenticazione, incluso SSLCIPH, CipherSpec.

Configurazione di un client transazionale esteso per CICS

Configurare un client transazionale esteso per l'utilizzo da parte di CICS aggiungendo una definizione della risorsa XAD a una regione CICS .

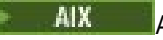

Aggiungere la definizione della risorsa XAD utilizzando il comando RDO (CICS resource definition online), **cicsadd**. La definizione della risorsa XAD specifica le seguenti informazioni:

- Una stringa xa_open
- Il nome percorso completo di un file di caricamento switch

Un file di caricamento switch viene fornito per l'utilizzo da parte di CICS su ognuna delle seguenti piattaforme:

-  AIX
-  Windows

Ogni file di caricamento switch contiene una funzione che restituisce un puntatore alla struttura di switch XA utilizzata per la registrazione dinamica, MQRMIXASwitchDynamic. Consultare [Tabella 5 a pagina 28](#) per il nome percorso completo di ciascun file di caricamento switch.

Tabella 5. I file di caricamento switch	
Piattaforma	File di caricamento switch
 AIX  Linux	MQ_INSTALLATION_PATH/lib/amqczsc
Windows	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll ¹

MQ_INSTALLATION_PATH rappresenta la directory di livello superiore in cui è installato IBM MQ .

Di seguito è riportato un esempio di definizione di risorsa XAD per i sistemi Windows :

```
cicsadd -c xad -r REGION1 WMQXA \
  ResourceDescription="IBM MQ queue manager MARS" \
  XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \
  SwitchLoadFile="C:\Program Files\IBM\MQ\bin\mqcc4swi.dll"
```

Per ulteriori informazioni sull'aggiunta di una definizione di risorsa XAD a una region CICS , consultare *CICS Administration Reference* e *CICS Administration Guide* per la propria piattaforma.

Tenere presente le seguenti informazioni sull'utilizzo di CICS con un client transazionale esteso:

- Puoi aggiungere solo una definizione di risorsa XAD per IBM MQ a una regione CICS . Ciò significa che solo un gestore code può essere associato a una region e tutte le applicazioni CICS eseguite nella region possono connettersi solo a quel gestore code. Se si desidera eseguire le applicazioni CICS che si connettono a un gestore code differente, è necessario eseguire le applicazioni in una regione diversa.
- Ogni server delle applicazioni in una regione richiama xa_open durante l'inizializzazione e l'avvio di un canale MQI per il gestore code associato alla regione. Ciò significa che il Gestore code deve essere avviato prima dell'avvio di un server delle applicazioni, altrimenti la chiamata xa_open non riesce. Tutte le applicazioni IBM MQ MQI client successivamente elaborate dal server delle applicazioni utilizzano lo stesso canale MQI.
- Quando viene avviato un canale MQI e non è presente alcuna uscita di protezione all'estremità client del canale, l'ID utente che passa dal sistema client alla connessione server MCA è cics . In determinate circostanze, il gestore code utilizza questo ID utente per i controlli di autorizzazione quando l'MCA di connessione del server tenta successivamente di accedere alle risorse del gestore code per conto di un'applicazione client. Se questo ID utente viene utilizzato per i controlli dell'autorità, è necessario assicurarsi che disponga dell'autorizzazione per accedere a tutte le risorse a cui deve accedere.

Per informazioni su quando il gestore code utilizza questo ID utente per i controlli di autorizzazione, consultare [Protezione](#).

- Le uscite di fine attività CICS fornite per l'utilizzo su sistemi client IBM MQ sono elencate in [Tabella 6](#) a pagina 29 . Le uscite vengono configurate nello stesso modo in cui vengono configurate le uscite corrispondenti per sistemi server IBM MQ . Per queste informazioni, quindi, consultare [Abilitazione delle uscite utente CICS](#).

Tabella 6. Uscite di terminazione attività CICS		
Piattaforma	Origine	Libreria
<ul style="list-style-type: none"> > AIX AIX > Linux Linux 	amqzscgx.c	amqzscg
Sistemi Windows Windows	amqzscgn.c	mqqc1415.dll

Configurazione di un client transazionale esteso per Tuxedo

Per configurare la definizione della risorsa XAD per l'utilizzo da parte di Tuxedo, aggiornare il file UBBCONFIG e la tabella del gestore risorse.

Per configurare la definizione della risorsa XAD per l'utilizzo da parte di Tuxedo, effettuare quanto segue:

- Nella sezione GROUPS del file Tuxedo UBBCONFIG per una applicazione, utilizzare il parametro **OPENINFO** per specificare una stringa xa_open. Per un esempio di come eseguire questa operazione, consultare il file UBBCONFIG di esempio, fornito per l'utilizzo con i programmi di esempio Tuxedo.

> **AIX** Sulle piattaforme seguenti, il nome del file è ubbstxcx.cfg:

– AIX

> **Windows** Windows, il nome del file è ubbstxcn.cfg.

- Nella voce per un gestore code nella tabella del gestore risorse Tuxedo, specificare il nome di una struttura switch XA e il nome percorso completo della libreria che contiene la struttura:

– > **AIX** In AIX, specificare udataobj/RM.

– **Windows** In Windows, specificare udataobj\rm.

Per un esempio di come eseguire questa operazione per ciascuna piattaforma, consultare [Esempi TUXEDO](#). Tuxedo supporta la registrazione dinamica di un gestore risorse e, pertanto, è possibile utilizzare MQRMIXASwitch o MQRMIXASwitchDynamic.

Windows Server Microsoft Transaction

Non è richiesta alcuna configurazione aggiuntiva prima di poter utilizzare Microsoft Transaction Server (MTS) come gestore transazioni. Tuttavia, ci sono alcuni punti da notare.

Tenere presenti le seguenti informazioni sull'utilizzo di MTS con il client transazionale esteso:

- Un'applicazione MTS avvia sempre un canale MQI quando si connette a un gestore code del server. MTS, nel ruolo di gestore transazioni, utilizza lo stesso canale MQI per comunicare con il gestore code.
- A seguito di un errore, MTS deve essere in grado di recuperare eventuali unità di lavoro incomplete. Per far ciò, MTS deve essere in grado di comunicare con qualsiasi gestore code che partecipava a un'unità di lavoro incompleta al momento dell'errore.

Quando un'applicazione MTS si connette a un gestore code del server e avvia un canale MQI, il client transazionale esteso estrae informazioni sufficienti dai parametri della chiamata MQCONN o MQCONNX per consentire il riavvio del canale in seguito a un errore, se necessario. Il client transazionale esteso trasmette le informazioni a MTS e MTS registra le informazioni nel relativo log.

Se l'applicazione MTS emette una chiamata MQCONN, queste informazioni sono semplicemente il nome del gestore code. Se l'applicazione MTS emette una chiamata MQCONNX e fornisce una struttura di definizione del canale, MQCD, le informazioni includono anche il nome del canale MQI, l'indirizzo di rete del gestore code del server e il protocollo di comunicazione per il canale.

In una situazione di recupero, MTS restituisce queste informazioni al client transazionale esteso, che utilizza per riavviare il canale MQI.

Se è necessario modificare le informazioni di configurazione, assicurarsi che tutte le unità di lavoro incomplete siano state risolte prima di apportare le modifiche. In alternativa, verificare che le modifiche di configurazione non influiscano sulla capacità del client transazionale esteso di riavviare un canale MQI utilizzando le informazioni registrate da MTS. Di seguito sono riportati esempi di tali modifiche di configurazione:

- Modifica del valore della variabile di ambiente MQSERVER
 - Modifica delle voci nella tabella di definizione del canale client (CCDT)
 - Eliminazione di una definizione di canale di connessione server
- Si notano le condizioni seguenti quando si utilizza un client transazionale esteso con MTS:
 - All'interno di un singolo thread, un'applicazione client può essere connessa a un solo gestore code alla volta.
 - Ogni thread di una applicazione client può connettersi a un gestore code differente.
 - Un'applicazione client non può utilizzare handle di connessione condivisi.

Definizione di canali MQI

Per creare un nuovo canale, devi creare **due** definizioni di canale, una per ogni estremità della connessione, utilizzando lo stesso nome di canale e tipi di canale compatibili. In questo caso, i tipi di canali sono *connessione server* e *connessione client*.

Canali definiti dall'utente

Quando il server non definisce automaticamente i canali, esistono due modi per creare le definizioni dei canali e fornire all'applicazione IBM MQ sulla macchina IBM MQ MQI client l'accesso al canale.

Questi due metodi sono descritti in dettaglio:

1. Creare una definizione di canale sul client IBM MQ e l'altra sul server.

Ciò si applica a qualsiasi combinazione di piattaforme IBM MQ MQI client e server. Utilizzarlo quando si inizia a utilizzare il sistema o per verificare la configurazione.

Consultare [“Creazione di definizioni di connessione server e di connessione client su piattaforme differenti”](#) a pagina 36 per i dettagli su come utilizzare questo metodo.

2. Creare entrambe le definizioni di canale sulla macchina server.

Utilizzare questo metodo quando si impostano più canali e macchine IBM MQ MQI client contemporaneamente.

Consultare [“Creazione di definizioni di connessioni server e client sul server”](#) a pagina 42 per i dettagli su come utilizzare questo metodo.

Canali definiti automaticamente

Multi

I prodotti IBM MQ su Multiplatforms includono una funzione che può creare automaticamente una definizione di canale sul server se non ne esiste una.

Se una richiesta di collegamento in entrata viene ricevuta da un client e non è possibile trovare una definizione di connessione server appropriata su tale gestore code, IBM MQ crea una definizione automaticamente e la aggiunge al gestore code. La definizione automatica è basata sulla definizione del canale di connessione server predefinito SYSTEM.AUTO.SVRCONN. È possibile abilitare la definizione automatica delle definizioni di connessione server aggiornando l'oggetto del gestore code utilizzando il comando ALTER QMGR con il parametro CHAD (o il comando PCF Modifica gestore code con il parametro ChannelAutoDef).

Concetti correlati

[“Funzione di controllo canale”](#) a pagina 235

La funzione di controllo del canale consente di definire, monitorare e controllare i canali.

ALW

Creazione e utilizzo di canali AMQP

Quando si installa il supporto IBM MQ per il componente del servizio AMQP nell'installazione di IBM MQ, è possibile eseguire i comandi MQSC IBM MQ (**runmqsc**) per definire, modificare, eliminare, avviare e arrestare un canale. È anche possibile visualizzare lo stato di un canale.

Prima di iniziare

Questa attività presuppone che sia installato il canale AMQP. A tale scopo, selezionare il componente AMQP SERVICE durante l'installazione di IBM MQ. Per ulteriori informazioni, segui il link per la tua piattaforma e trova la riga della tabella per "AMQP Service":

- ▶ **AIX** [Componenti IBM MQ per sistemi AIX](#)
- ▶ **Linux** [IBM MQ per sistemi Linux](#)
- ▶ **Linux** [IBM MQ Debian per sistemi Linux Ubuntu](#)
- ▶ **Windows** [Funzioni IBM MQ per sistemi Windows](#)

Nota: Vedi [Restarting the IBM MQ service for AMQP](#) per un esempio di componente SERVICE e ulteriori informazioni se il tuo servizio AMQP smette di funzionare correttamente.

Questa attività presuppone anche che si disponga di un gestore code esistente.

Per stabilire una connessione di prova al gestore code, è possibile utilizzare qualsiasi client AMQP che implementa il protocollo OASIS AMQP 1.0, come i client MQ Light e Apache Qpid come Apache Qpid Proton e Apache Qpid JMS.

Da IBM MQ 9.3.0, è possibile utilizzare solo il canale predefinito, SYSTEM.DEF.AMQP, per verificare le connessioni MQ Light al gestore code. La seguente procedura utilizza il canale predefinito.

Questa attività si basa sul client MQ Light Node.js . Tuttavia, i passaggi relativi al gestore code IBM MQ sono gli stessi per qualsiasi client.

Nota: I canali AMQP non supportano i servizi AMQP definiti dall'utente. I canali AMQP supportano solo il sistema predefinito SYSTEM.AMQP.SERVICE . È possibile definire solo un'istanza di questo servizio per ciascun gestore code.

Procedura

1. Avviare **runmqsc** dalla directory `mqinstall/bin/` :

```
runmqsc QMNAME
```

2. (Necessario solo se il gestore code è `V 9.4.0` da IBM MQ 9.4.0 o se è IBM MQ 9.0.4 o precedente). Verificare che la funzione AMQP sia installata e che funzioni correttamente. Utilizzare il comando **START SERVICE** per avviare il servizio IBM MQ , che controlla la JVM:

```
START SERVICE(SYSTEM.AMQP.SERVICE)
```

Nota: Da IBM MQ 9.4.0 il SYSTEM.AMQP.SERVICE ha il proprio attributo **CONTROL** impostato su MANUAL. Ciò impedisce l'avvio del servizio all'avvio del gestore code. L'impostazione della propriet ... **CONTROL** su QMGR, viene avviata automaticamente all'avvio del gestore code.

Da IBM MQ 9.1 a IBM MQ 9.3, SYSTEM.AMQP.SERVICE ha l'attributo **CONTROL** impostato su QMGR.

3. Impostare l'ID utente MCAUSER .

Quando un client AMQP si connette a un canale, il canale specifica un ID utente MCAUSER , che viene utilizzato sulle connessioni al gestore code. Il valore predefinito di MCAUSER è vuoto. Prima che qualsiasi client AMQP possa connettersi al gestore code, è necessario specificare un valore MCAUSER , che deve essere un utente IBM MQ valido autorizzato a pubblicare e sottoscrivere argomenti IBM MQ .

- a) Utilizzare il comando **ALTER CHANNEL** per impostare l'ID utente MCAUSER :

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) MCAUSER(Usert ID)
```

- b) Utilizzare i due comandi **setmqaut** riportati di seguito per autorizzare l'ID utente MCAUSER a pubblicare e sottoscrivere argomenti:

```
setmqaut -m QMNAME -t topic -n SYSTEM.BASE.TOPIC -p MCAUSER  
-all +pub +sub
```

e

```
setmqaut -m QMNAME -t qmgr -p MCAUSER -all +connect
```

Se il canale è in esecuzione mentre l'ID utente MCAUSER viene aggiunto o modificato, è necessario arrestare e riavviare il canale.

Nota: Se l'ID utente MCAUSER non è impostato oppure se l'ID utente MCAUSER non è autorizzato a pubblicare o sottoscrivere argomenti IBM MQ , si riceverà un messaggio di errore nel client AMQP.

4. Utilizzare il comando **START CHANNEL** per avviare il sistema SYSTEM.DEF.AMQP :

```
START CHANNEL(SYSTEM.DEF.AMQP)
```

5. Se si desidera controllare lo stato del canale, utilizzare il comando **DISPLAY CHSTATUS** :


```
DISPLAY CHSTATUS(SYSTEM.DEF.AMQP) CHLTYPE(AMQP)
```

Quando il canale viene eseguito correttamente, STATUS(RUNNING) viene visualizzato nell'output del comando.

6. Modificare la porta predefinita.

La porta predefinita per le connessioni AMQP 1.0 è 5672. Se si sta già utilizzando la porta 5672, che è possibile se è stato precedentemente installato MQ Light, è necessario modificare la porta utilizzata dal canale AMQP. Utilizzare il comando **ALTER CHANNEL** per modificare la porta:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) PORT(NEW PORT NUMBER)
```

7. Se non si desidera bloccare o filtrare le connessioni al canale AMQP utilizzando le regole di autenticazione di canale (CHLAUTH), disabilitare l'autenticazione di canale sul gestore code nel modo seguente:

```
alter qmgr chlauth(disabled)
```

Si consiglia di non disabilitare l'autenticazione della connessione su un gestore code di produzione. È necessario disabilitare solo l'autenticazione della connessione in un ambiente di sviluppo.

In alternativa, configurare le regole di autenticazione del canale del gestore code per consentire connessioni specifiche al canale AMQP.

8. Opzionale: Se si desidera abilitare la codifica SSL/TLS sul canale, utilizzando il repository delle chiavi configurato per il gestore code, è necessario impostare l'attributo SSLCIPH per il canale su una specifica di cifratura appropriata. Per impostazione predefinita, la specifica di cifratura è vuota, il che significa che la codifica SSL/TLS non viene utilizzata nel canale. Utilizzare il comando **ALTER CHANNEL** per impostare una specifica di cifratura. Ad esempio:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCIPH(CIPHER SPECIFICATION)
```

Inoltre, ci sono una serie di altre opzioni di configurazione del canale associate alla crittografia SSL/TLS che puoi impostare come segue:

- Per impostazione predefinita, il certificato nel repository delle chiavi del gestore code con l'etichetta corrispondente all'attributo **CERTLABL** del gestore code è il nome utilizzato dalla codifica SSL/TLS per il canale. È possibile selezionare un certificato differente impostando **CERTLABL**. Utilizzare il comando **ALTER CHANNEL** per specificare l'etichetta per il certificato richiesto:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) CERTLABL(CERTIFICATE LABEL)
```

- È possibile configurare il canale per richiedere un certificato dalle connessioni client SSL/TLS. È possibile selezionare se un certificato è richiesto da una connessione client SSL/TLS impostando l'attributo **SSLCAUTH**. Utilizzare il comando **ALTER CHANNEL** per impostare se un certificato è richiesto da una connessione client SSL/TLS. Ad esempio:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCAUTH(REQUIRED OR OPTIONAL)
```

- Se si imposta l'attributo **SSLCAUTH** su REQUIRED, è possibile controllare il DN (Distinguished Name) del certificato dal client. Per controllare il DN del certificato dal client, impostare l'attributo **SSLPEER**. Utilizzare il comando **ALTER CHANNEL** per controllare il DN (Distinguished Name) del certificato dal client. Ad esempio:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLPEER (DN SPECIFICATION)
```

In alternativa, è possibile utilizzare i record di autenticazione di canale anche per consentire o bloccare le connessioni perché questo metodo offre una maggiore granularità rispetto all'utilizzo

dell'attributo **SSLPEER** . Per ulteriori informazioni sull'impostazione di **SSLPEER** e sull'utilizzo dei record di autenticazione di canale come alternativa, vedi [Peer SSL](#).

9. Installa il client MQ Light Node.js immettendo il seguente comando:

```
npm install mqlight
```

10. Passare alla directory `node_modules/mqlight/samples` ed eseguire l'applicazione del destinatario di esempio:

- Se si sta utilizzando il numero di porta predefinito, è possibile eseguire l'applicazione del destinatario di esempio:

```
node recv.js
```

- Se il canale AMQP è stato configurato per utilizzare un numero di porta differente, è possibile eseguire l'applicazione del destinatario di esempio con un parametro per specificare il nuovo numero di porta:

```
node recv.js -s amqp://localhost:6789
```

Una connessione corretta al canale predefinito visualizza il messaggio seguente:

```
Connected to amqp://localhost:5672 using client-id recv_e79c55d
Subscribed to pattern: public
```

L'applicazione è ora connessa al gestore code ed è in attesa di ricevere messaggi. È sottoscritto all'argomento `public`.

Nota: Il `client-id` viene generato automaticamente, a meno che non ne venga specificato uno utilizzando il parametro `-i`.

11. In una nuova finestra di comandi, passare alla directory `node_modules/mqlight/samples` ed eseguire l'applicazione mittente di esempio eseguendo il seguente comando:

```
node send.js
```

Nella finestra comandi per l'applicazione ricevente, viene visualizzato il messaggio `Hello World`.

12. Utilizzare l'esempio **AMQSSUB** IBM MQ per ricevere un MQ Light messaggio di esempio.

Su Linux e Windows, l'esempio è disponibile nelle seguenti ubicazioni:

- **Linux** `mqinstall/samp/bin` su Linux.
- **Windows** `mqinstall/Tools\c\Samples\Bin` su Windows.

a) Eseguire l'esempio immettendo il seguente comando:

```
amqssub public QM-name.
```

b) Inviare un messaggio all'applicazione IBM MQ eseguendo nuovamente il seguente comando:

```
node send.js
```

13. Utilizzare il comando **DEFINE CHANNEL** per creare più canali AMQP:

```
DEFINE CHANNEL(MY.AMQP.CHANNEL) CHLTYPE(AMQP) PORT(2345)
```

Quando si definisce un canale, è necessario avviarlo manualmente, utilizzando il comando **START CHANNEL** :

```
START CHANNEL(MY.AMQP.CHANNEL)
```

Per controllare che il canale sia in esecuzione correttamente, è possibile eseguire l'applicazione ricevente di esempio, specificando la porta del nuovo canale:

```
node recv.js -s amqp://localhost:2345
```

Operazioni successive

È possibile utilizzare i comandi riportati di seguito per visualizzare le connessioni IBM MQ , arrestare il canale ed eliminare il canale:

DISPLAY CONN(*) TYPE(CONN) WHERE (CHANNEL EQ SYSTEM.DEF.AMQP)

Visualizza la connessione IBM MQ effettuata dal canale AMQP sul gestore code.

DISPLAY CHSTATUS(*) CHLTYPE(AMQP) CLIENTID(*) ALL

Visualizza un elenco di client AMQP connessi al canale specificato.

STOP CHANNEL (MY.AMQP.CHANNEL)

Arresta un canale AMQP e chiude la porta su cui è in ascolto.

DELETE CHANNEL (MY.AMQP.CHANNEL)

Elimina tutti i canali creati.

Nota: Non eliminare il canale predefinito SYSTEM.DEF.AMQPAMQP.

È possibile determinare se la funzionalità AMQP è installata nell'installazione di IBM MQ e se ad essa è associato un gestore code, utilizzando **runmqsc** o PCF:

- Utilizzando **runmqsc**, visualizzare gli attributi del gestore code e controllare AMQPCAP (YES).
- Utilizzando PCF, utilizzare il comando **MQCMD_INQUIRE_Q_MGR** e verificare il valore di MQIA_AMQP_CAPABILITY.

Attività correlate

[Sviluppo di applicazioni client AMQP](#)

[Protezione dei client AMQP](#)

Riferimenti correlati

[strmqm](#)

ALW Rimozione del canale AMQP dai gestori code

È possibile eliminare il canale AMQP dai gestori code rimuovendo cartelle dalla directory di installazione.

Procedura

1. Chiudere il gestore code.
2. Rimuovere il supporto IBM MQ per le API del componente del servizio AMQP:

- **AIX** Su AIX, eseguire il seguente comando:

```
installp -u mqm.amqp.rte
```

- **Linux** Su Linux, rimuovere l'RPM AMQP. Se si è riassembleto l'RPM prima di installarlo, specificare il nome dell'RPM riassembleto.

```
rpm -e MQSeriesAMQP
```

- **Windows** Su Windows, eliminare la cartella amqp dall'installazione di IBM MQ . Verificare che nessun altro file o cartella nel percorso di installazione di IBM MQ venga rimosso.

3. Riavviare il gestore code.

Attività correlate

[Sviluppo di applicazioni client AMQP](#)

[Protezione dei client AMQP](#)

ALW File di log del canale AMQP

I file di log per i canali AMQP sono memorizzati nella stessa directory di dati IBM MQ dei file di log IBM MQ .

La directory dei dati predefinita su Windows è C:\ProgramData\IBM\MQ.

La directory dei dati predefinita su Linux è /var/mqm.

Il canale AMQP scrive le informazioni di log nei seguenti file di log, disponibili nella directory di dati IBM MQ :

- amqp.stdout, scritto nella cartella qmgrs/QM-name .
- amqp.stderr, scritto nella cartella qmgrs/QM-name .
- amqp_*.log , scritto nella cartella qmgrs/QM-name/errors .
- **V9.4.0** **V9.4.0** amqp_*.json , scritto nella cartella qmgrs/QM-name/errors .

Se un client MQ Light riceve un errore di autenticazione o di autorizzazione, l'amministratore può trovare informazioni dettagliate sul motivo dell'errore di sicurezza nel file amqp_0.log ,

V9.4.0 **V9.4.0** amqp_0.json filee nei file MQ AMQERR*.log .

Tutti i file FDC vengono creati come file AMQP*.FDC , scritti nella cartella *data-directory/errors* .

Alcuni file di configurazione vengono scritti nella directory qmgrs/QM-name/amqp .

V9.4.0 **V9.4.0** I log formattati JSON in AMQP sono facoltativi e devono essere abilitati manualmente. A tale scopo, modificare i [log AMQP, log di errori e file di configurazione](#).

Concetti correlati

[Log degli errori su AIX, Linux, and Windows](#)

Attività correlate

[Sviluppo di applicazioni client AMQP](#)

[Protezione dei client AMQP](#)

[Abilitazione dei log formattati JSON per AMQP](#)

Creazione di definizioni di connessione server e di connessione client su piattaforme differenti

È possibile creare ogni definizione di canale sul computer a cui si applica. Tuttavia, esistono delle limitazioni su come è possibile creare definizioni di canale su un computer client.

Informazioni su questa attività

Su tutte le piattaforme, è possibile utilizzare i comandi IBM MQ Script (MQSC), i comandi PCF (programmable command format) o IBM MQ Explorer per definire un canale di connessione server sulla macchina server.

z/OS Su z/OS è anche possibile utilizzare i pannelli Operazione e Controllo.

IBM i Su IBM i è anche possibile utilizzare l'interfaccia del pannello.

Poiché i comandi MQSC non sono disponibili su una macchina in cui IBM MQ è stato installato solo come IBM MQ MQI client, è necessario utilizzare diversi metodi di definizione di un canale di connessione client sulla macchina client.

Le seguenti considerazioni si applicano quando **runmqsc**:

- È possibile specificare il parametro **-c** e, facoltativamente, il parametro **-u** per connettersi **runmqsc** come client al gestore code che si desidera gestire.
- Se si utilizza il parametro **-u** per specificare un ID utente, viene richiesta una password corrispondente.
- Se il record CONNAUTH AUTHINFO è stato configurato con CHCKLOCL (REQUIRED) o CHCKLOCL (REQDADM), è necessario utilizzare il parametro **-u** altrimenti non sarà possibile gestire il gestore code con **runmqsc**.

Procedura

- Per definire un canale di connessione server sul server, consultare [“Definizione di un canale di connessione server sul server”](#) a pagina 37.
- Per creare un canale di collegamento client su IBM MQ MQI client utilizzando la variabile di ambiente **MQSERVER**, consultare [“Creazione di un canale di connessione client su IBM MQ MQI client utilizzando MQSERVER”](#) a pagina 38.
- Per creare un canale di connessione client su IBM MQ MQI client utilizzando la struttura MQCNO su una chiamata MQCONN, consultare [“Creazione di un canale di connessione client su IBM MQ MQI client utilizzando MQCNO”](#) a pagina 42.

Definizione di un canale di connessione server sul server

Avviare MQSC, se necessario, quindi definire il canale di connessione server.

Procedura

1. Opzionale: Se si utilizza un server Multiplatforms, prima creare e avviare un gestore code, quindi avviare i comandi MQSC.
 - a) Creare un gestore code, denominato QM1, ad esempio:

```
crtmqm QM1
```

- b) Avviare il gestore code:

```
strmqm QM1
```

- c) Comandi di avvio MQSC:

```
runmqsc QM1
```

2. Definire un canale con il nome scelto e un tipo di canale *server - connection*.

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_1')
```

Questa definizione di canale è associata al gestore code in esecuzione sul server.

3. Utilizzare il seguente comando per consentire l'accesso di connessione in entrata al gestore code:

```
SET CHLAUTH(CHAN1) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- Dove SET CHLAUTH utilizza il nome del canale definito nel passo precedente.
- Dove *'indirizzo IP'* è l'indirizzo IP del client.

- Dove *'userid'* è l'ID che si desidera fornire al canale per il controllo dell'accesso alle code di destinazione. Questo campo è sensibile al maiuscolo / minuscolo.

È possibile scegliere di identificare la connessione in entrata utilizzando un numero di attributi differenti. L'esempio utilizza l'indirizzo IP. Gli attributi alternativi includono l'ID utente client e il DN (Distinguished Name) oggetto TLS. Per ulteriori informazioni, consultare [Record di autenticazione di canale](#)

Creazione di un canale di connessione client su IBM MQ MQI client utilizzando MQSERVER

È possibile definire un canale di connessione client su una workstation client utilizzando la variabile di ambiente **MQSERVER**.

Informazioni su questa attività

È possibile utilizzare la variabile di ambiente **MQSERVER** per specificare una semplice definizione di canale di connessione client. È semplice nel senso che è possibile specificare solo alcuni attributi del canale utilizzando questo metodo.

Se si utilizza la variabile di ambiente **MQSERVER** per definire il canale tra la macchina IBM MQ MQI client e una macchina server, questo è l'unico canale disponibile per l'applicazione e non viene fatto alcun riferimento alla CCDT (client channel definition table).

Se la richiesta MQCONN o MQCONNX specifica un gestore code diverso da quello a cui è connesso il listener o se il **MQSERVER** parametro *TransportType* non viene riconosciuto, la richiesta MQCONN o MQCONNX ha esito negativo con codice di ritorno MQRC_Q_MGR_NAME_ERROR.

Linux **AIX** Su AIX and Linux, è possibile definire **MQSERVER** come in uno dei seguenti esempi:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

Tutte le richieste MQCONN o MQCONNX tentano quindi di utilizzare il canale definito a meno che non sia stato fatto riferimento a una struttura MQCD dalla struttura MQCNO fornita a MQCONNX, nel qual caso il canale specificato dalla struttura MQCD ha la priorità su qualsiasi valore specificato dalla variabile di ambiente **MQSERVER**.

La variabile di ambiente **MQSERVER** ha la priorità su qualsiasi definizione di canale client indicata dalle variabili di ambiente **MQCHLLIB** e **MQCHLTAB**.

Procedura



- In base alla piattaforma, utilizzare uno dei seguenti comandi per specificare la definizione di canale con **MQSERVER**.

– **Windows** Su Windows, specificare una definizione di canale semplice come segue:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

Ad esempio:


```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

-   Su AIX and Linux, specificare una definizione di canale semplice come segue:

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

Ad esempio:

```
SET MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```



-  Su IBM i, specificare una definizione di canale semplice come segue:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('ChannelName/TransportType/ConnectionName')
```

Ad esempio:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

Note:

- *ChannelName* deve essere lo stesso nome definito sul server. Non può contenere il carattere barra (/) perché questo carattere viene utilizzato per separare il nome del canale, il tipo di trasporto e il nome della connessione. Quando la variabile di ambiente **MQSERVER** viene utilizzata per definire un canale client, viene utilizzata una lunghezza massima del messaggio (**MAXMSGL**) di 100 MB. Pertanto, la dimensione massima del messaggio in vigore per il canale è il valore specificato nel canale SVRCONN sul server.
- Il *TransportType* può essere uno tra LU62, TCP, NETBIOS, SPX, a seconda della piattaforma client IBM MQ.
-   Su AIX and Linux, *TransportType* è sensibile al maiuscolo / minuscolo e deve essere maiuscolo. Una chiamata MQCONN o MQCONNX restituisce 2058 se il tipo di trasporto non è riconosciuto.
- *ConnectionName* è il nome del server come definito nel protocollo di comunicazione (*TransportType*). Deve essere un nome di rete completo, ad esempio AMACHINE.ACOMPANY.COM(1414).
- *ConnectionName* può essere un elenco separato da virgole di nomi di connessione. I nomi di connessione nell'elenco vengono utilizzati in modo simile a più connessioni in una tabella di connessioni client. L'elenco dei nomi delle connessioni potrebbe essere utilizzato come alternativa ai gruppi di gestori code per specificare più collegamenti per il client da provare. Se si sta configurando un gestore code a più istanze, è possibile utilizzare un elenco di nomi di connessione per specificare diverse istanze del gestore code.
- Per annullare **MQSERVER** e tornare alla tabella di definizione del canale client indicata da **MQCHLLIB** e **MQCHLTAB**, immettere il seguente comando:

-   Su AIX and Linux:

```
unset MQSERVER
```

-  Su Windows:

```
SET MQSERVER=
```

Esempio

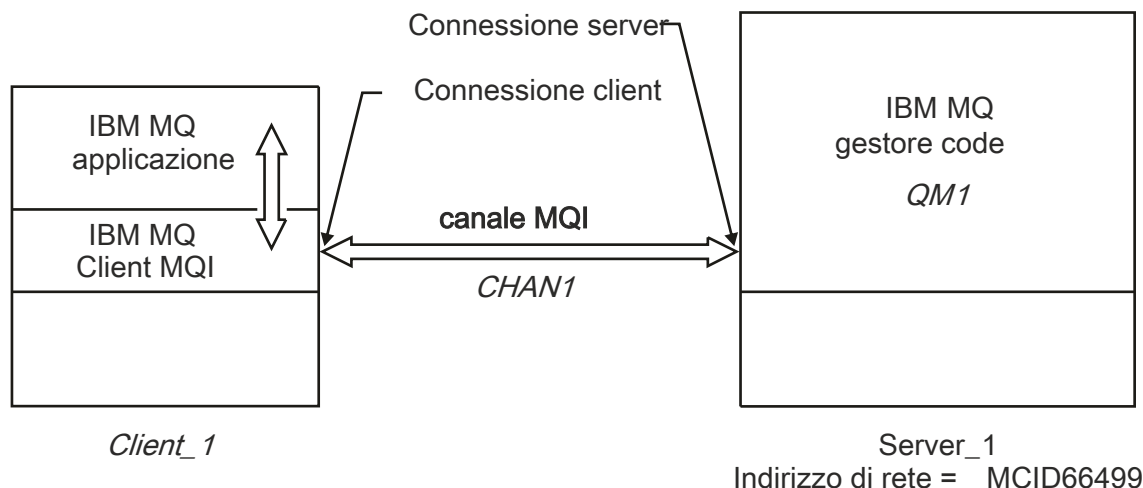


Figura 1. Esempio di una definizione di canale semplice

Per creare la definizione di canale semplice mostrata nella [Figura 1 a pagina 40](#), utilizzare i seguenti comandi:

- **Linux** **AIX** Su AIX and Linux:

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

- **Windows** Su Windows:

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

Nota: Per informazioni su come modificare il numero di porta TCP/IP, consultare [“Modifica della porta predefinita TCP/IP”](#) a pagina 41.

Di seguito sono riportati altri esempi di definizioni di canali semplici:

- **Windows** Su Windows:

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56  
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- **Linux** **AIX** Su AIX and Linux:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

dove BOX99 è la LU 6.2 ConnectionName.

- **IBM i** Su IBM i:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('CHANNEL1/TCP/9.20.4.56(1416)')
```

Su IBM MQ MQI client, tutte le richieste **MQCONN** o **MQCONNX** tentano quindi di utilizzare il canale definito, a meno che il canale non venga sovrascritto in una struttura MQCD a cui si fa riferimento dalla struttura MQCNO fornita a **MQCONNX**.

Attività correlate

“Utilizzo delle variabili d'ambiente IBM MQ” a pagina 64

È possibile utilizzare i comandi per visualizzare le impostazioni correnti o per reimpostare i valori delle variabili di ambiente IBM MQ .

“Creazione di un canale di connessione client su IBM MQ MQI client utilizzando MQCNO” a pagina 42

È possibile definire un canale di connessione client sulla stazione di lavoro client utilizzando la struttura MQCNO su una chiamata MQCONN.

Modifica della porta predefinita TCP/IP

Per impostazione predefinita, per TCP/IP, IBM MQ presuppone che il canale sia connesso alla porta 1414. Se necessario, è possibile modificare il valore predefinito.

Informazioni su questa attività

È possibile modificare il numero di porta utilizzando una delle seguenti tre opzioni:

- Utilizzando la variabile di ambiente **MQSERVER** .
- Modificando il file `mqclient.ini` .
- Aggiungendo IBM MQ al file dei servizi.

Procedura

- Per modificare il numero di porta utilizzando la variabile di ambiente **MQSERVER** , aggiungere il numero di porta tra parentesi come ultima parte di *ConnectionName*, ad esempio:

–   Su AIX and Linux:

```
export MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
```

–  Su Windows:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- Per modificare il numero di porta utilizzando il file `mq.ini` , aggiungere il numero di porta al nome protocollo, ad esempio:

```
TCP:  
port=2001
```

- Per cambiare il numero di porta aggiungendo IBM MQ al file dei servizi, completare la procedura descritta in “Utilizzo del listener TCP/IP su AIX and Linux” a pagina 278.

Modifica del socket predefinito SPX

Per impostazione predefinita, per SPX IBM MQ presuppone che il canale sia connesso al socket 5E86. Se necessario, è possibile modificare il valore predefinito.

Informazioni su questa attività

È possibile modificare il numero di porta utilizzando una delle seguenti opzioni:

- Utilizzando la variabile di ambiente **MQSERVER** .

Per connessioni SPX, specificare *ConnectionName* e *socket* nel formato `network.node(socket)`. Se il client e il server IBM MQ si trovano sulla stessa rete, non è necessario specificare la rete. Se si utilizza il socket predefinito, non è necessario specificarlo.

- Modificando la stanza SPX del `mqclient.ini` file.Changing nel file `qm.ini` .

Procedura

- Per modificare il numero di porta per una connessione SPX utilizzando la variabile di ambiente **MQSERVER**, specificare *ConnectionName* ed il socket nel formato `network.node(socket)` come mostrato nel seguente esempio:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)
```

Nota: Se il client e il server IBM MQ si trovano sulla stessa rete, non è necessario specificare la rete. Se si utilizza il socket predefinito, non è necessario specificare il socket.

- Per modificare il numero di porta utilizzando il file `qm.ini`, aggiungere il numero di porta al nome protocollo, ad esempio:

```
SPX:  
socket=5E87
```

Creazione di un canale di connessione client su IBM MQ MQI client utilizzando MQCNO

È possibile definire un canale di connessione client sulla stazione di lavoro client utilizzando la struttura MQCNO su una chiamata MQCONNX.

Informazioni su questa attività

Un'applicazione IBM MQ MQI client può utilizzare la struttura delle opzioni di connessione, MQCNO, su una chiamata **MQCONNX** per fare riferimento a una struttura di definizioni di canale, MQCD, contenente la definizione di un canale di connessione client.

In questo modo, l'applicazione client può specificare gli attributi **ChannelName**, **TransportType** e **ConnectionName** di un canale al runtime, consentendo all'applicazione client di connettersi a più gestori code del server contemporaneamente.

Notare che se si definisce un canale utilizzando la variabile di ambiente **MQSERVER**, non è possibile specificare gli attributi **ChannelName**, **TransportType** e **ConnectionName** al runtime.

Un'applicazione client può anche specificare gli attributi di un canale come **MaxMsgLength** e **SecurityExit**. La specifica di tali attributi consente all'applicazione client di specificare i valori per gli attributi che non sono i valori predefiniti e consente ai programmi di uscita del canale di essere richiamati all'estremità client di un canale MQI.

Se un canale utilizza TLS (Transport Layer Security), un'applicazione client può fornire anche informazioni relative a TLS nella struttura MQCD. Ulteriori informazioni relative a TLS possono essere fornite nella struttura delle opzioni di configurazione TLS, MQSCO, a cui fa riferimento anche la struttura MQCNO in una chiamata **MQCONNX**.

Per ulteriori informazioni sulle strutture MQCNO, MQCD e MQSCO, consultare [MQCNO](#), [MQCD](#) e [MQSCO](#).

Nota: Il programma di esempio per MQCONNX è denominato **amqscnxc**. Un altro programma di esempio denominato **amqssslc** dimostra l'utilizzo della struttura MQSCO.

Attività correlate

[“Creazione di un canale di connessione client su IBM MQ MQI client utilizzando MQSERVER” a pagina 38](#)





È possibile definire un canale di connessione client su una workstation client utilizzando la variabile di ambiente **MQSERVER**.

Creazione di definizioni di connessioni server e client sul server

È possibile creare entrambe le definizioni sul server, quindi rendere la definizione di connessione client disponibile per il client.

Informazioni su questa attività

Definire prima un canale di connessione server e quindi definire un canale di connessione client:

- Su tutte le piattaforme, è possibile utilizzare i comandi IBM MQ Script (MQSC), i comandi PCF (programmable command format) per definire un canale di connessione server sulla macchina server.
-   In Linux e Windows, è possibile utilizzare anche IBM MQ Explorer.
-  Su z/OS, è anche possibile utilizzare i pannelli Operazione e Controllo.
-  Su IBM i è anche possibile utilizzare l'interfaccia del pannello.

Le definizioni di canale di connessione client create sul server vengono rese disponibili ai client utilizzando una CCDT (client channel definition table).

Procedura

1. Per definire un canale di connessione server, consultare [“Definizione del canale di connessione server sul server”](#) a pagina 56.
2. Per definire un canale di connessione client, consultare [“Definizione del canale di connessione client nel server”](#) a pagina 57.

Attività correlate

[“Configurazione di un formato binario CCDT”](#) a pagina 44

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

[“Definizione del canale di connessione server sul server”](#) a pagina 56

Creare una definizione di canale di connessione server per il gestore code.

[“Definizione del canale di connessione client nel server”](#) a pagina 57

Una volta definito il canale di connessione server, è possibile definire il corrispondente canale di connessione client.

[“Accesso alle definizioni del canale di connessione client”](#) a pagina 58

È possibile rendere la tabella di definizione del canale client (CCDT) disponibile per le applicazioni client copiandola o condividendola, quindi specificarne l'ubicazione e il nome sul computer client. È anche possibile individuare una tabella di definizione del canale client (CCDT) tramite un URL.

Configurazione delle tabelle di definizione del canale client


Una tabella di definizione del canale client (CCDT) definisce i canali di connessione client e i loro attributi. I client leggono questo file per determinare a quali gestori code connettersi. Il file CCDT può essere in formato JSON o binario.

Informazioni su questa attività

Il gestore code non legge il file CCDT. Viene utilizzato solo per fornire definizioni di canale e informazioni di autenticazione ai client.

Un formato binario CCDT viene creato automaticamente quando viene creato un gestore code. Le definizioni di canale client memorizzate in questa tabella vengono aggiornate utilizzando solo il comando **runmqsc**.

Un formato JSON CCDT è un file di testo semplice con estensione .json. Creare e aggiornare manualmente questa tabella, che è meno restrittiva rispetto all'uso del comando **runmqsc**.

 I client z/OS JMS in esecuzione in un server delle applicazioni utilizzano una CCDT per fare riferimento ai dettagli di connessione del gestore code remoto. Da IBM MQ for z/OS 9.1, IBM MQ

Advanced for z/OS Value Unit Edition consente ai client JMS di connettersi in remoto ai gestori code su altre LPAR z/OS . Pertanto questi client possono utilizzare anche i CCDT.

Per facilitare la configurazione dei CCDT per l'utilizzo dei client, scegliere tra le attività riportate di seguito:

Procedura

- [“Configurazione di un formato binario CCDT” a pagina 44](#)
- [“Configurazione di un formato JSON CCDT” a pagina 46](#)
- [“Ubicazioni per la CCDT” a pagina 53](#)
- [“Accesso URL alla CCDT” a pagina 54](#)

Concetti correlati

Client MQI: [CCDT \(Client Channel Definition Table\)](#)

Attività correlate

[“Configurazione di un cluster uniforme” a pagina 425](#)

I cluster uniformi consentono alle applicazioni di essere progettate per la scalabilità e la disponibilità e possono connettersi a qualsiasi gestore code all'interno di tale cluster uniforme.

Configurazione di un formato binario CCDT

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

Prima di iniziare

È inoltre possibile creare una CCDT in formato JSON (JavaScript Object Notation) e l'uso di questo formato alternativo presenta alcuni vantaggi rispetto all'utilizzo di CCDT binario. Consultare [“Configurazione di un formato JSON CCDT” a pagina 46](#).

I client su tutte le piattaforme possono visualizzare e utilizzare i CCDT. Tuttavia, il CCDT binario può essere creato e modificato solo in IBM MQ for Multiplatforms.

Informazioni su questa attività

 Su [Multiplatforms](#):

- Un CCDT binario viene creato automaticamente nella directory @ipcc sotto la directory dei dati per il gestore code.
- Oltre ad essere creata automaticamente, la CCDT binaria associata ad un gestore code viene mantenuta sincronizzata con le definizioni oggetto. Quando si definisce, si modifica o si elimina un oggetto del canale client, sia la definizione dell'oggetto del gestore code che la voce in CCDT vengono aggiornate come parte della stessa operazione.

Note:

- La progettazione del file IBM MQ CCDT è che il file CCDT viene ridotto, solo dopo che tutti i canali di connessione client definiti dall'utente sono stati effettivamente definiti. Quando un canale di connessione client viene eliminato, viene appena contrassegnato come eliminato nel file CCDT, ma non viene fisicamente rimosso.
- Per forzare la riduzione del file CCDT, dopo aver eliminato uno o più canali di connessione client, immettere il seguente comando:

```
rcrmqobj -m QM80 -t clchltab
```

- Utilizzare il comando **runmqsc** per modificare l'ubicazione e il contenuto della CCDT binaria.

I client su tutte le piattaforme possono visualizzare e utilizzare una CCDT binaria.

Procedura

• **Multi**

Creare un CCDT binario predefinito.

Su Multipiattaforme, viene creata una CCDT binaria predefinita denominata AMQCLCHL . TAB quando si crea un gestore code.

Per impostazione predefinita, AMQCLCHL.TAB si trova nella seguente directory su un server:

– **IBM i** Su IBM i, nell'IFS:

```
/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/&ipcc
```

– **Linux** **AIX** Su sistemi AIX and Linux:

```
/prefix/qmgrs/QUEUEMANAGERNAME/@ipcc
```

il nome della directory a cui fa riferimento *QUEUEMANAGERNAME* è sensibile al maiuscolo / minuscolo sui sistemi AIX and Linux . Il nome della directory potrebbe non essere lo stesso del nome del gestore code, se il nome del gestore code contiene caratteri speciali.

– **Windows** Su Windows:

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEUEMANAGERNAME\@ipcc
```

dove *MQ_INSTALLATION_PATH* rappresenta la directory di alto livello in cui è installato IBM MQ .

Tuttavia, è possibile che si sia scelto di utilizzare una directory differente per i dati del gestore code. È possibile specificare il parametro **-md DataPath** quando si utilizza il comando **crtmqm** . In tal caso, AMQCLCHL . TAB si trova nella directory @ipcc del *DataPath* specificato.

• Individuare la CCDT:

- Sul computer client
- In una posizione condivisa da più di un cliente
- Sul server come file condiviso

Consultare [“Ubicazioni per la CCDT”](#) a pagina 53.

a) Creare un CCDT binario direttamente su una macchina client.

- Utilizzare il comando `runmqsc` con il parametro **-n** .
- La CCDT viene creata nell'ubicazione indicata da **MQCHLLIB** e con il nome file indicato da **MQCHLTAB**, che è AMQCLCHL . TAB per default.
- **Importante:** se si specifica il parametro **-n** , non è necessario specificare altri parametri.

b) Modificare l'ubicazione.

È possibile modificare il percorso della CCDT impostando **MQCHLLIB**. Tenere presente che, se si dispone di più gestori code sullo stesso server, essi condividono la stessa ubicazione CCDT.

• Accedi alla CCDT

È possibile accedere alla CCDT:

- In remoto da un file, ftp o URL http, definendo la variabile di ambiente **MQCCDTURL** .
- Localmente impostando le variabili di ambiente **MQCHLLIB** e **MQCHLTAB** .
- Localmente definendo gli attributi **ChannelDefinitionDirectory** e **ChannelDefinitionFile** della stanza CHANNELS nel file di configurazione del client.

Consultare [“Ubicazioni per la CCDT”](#) a pagina 53 per vari esempi.

• Visualizzare o modificare il contenuto CCDT.

È possibile visualizzare il contenuto CCDT con il comando `runmqsc` :

1. Impostare le variabili di ambiente su [Accesso a CCDT](#)
2. Esegui il comando `runmqsc -n`
3. Esegui il comando `DISPLAY CHANNEL (*)`, ad esempio

Multi Su Multiplatforms, è anche possibile modificare il contenuto CCDT binario utilizzando il comando `runmqsc`. Ogni voce di una CCDT rappresenta una connessione client a un gestore code specifico. Viene aggiunta una nuova voce quando si definisce un canale di connessione client utilizzando il comando **DEFINE CHANNEL**, e la voce viene aggiornata quando si modificano i canali di connessione client utilizzando il comando **ALTER CHANNEL**. Per ulteriori esempi di utilizzo del comando, consultare [runmqsc](#).

- Fornire ai client le informazioni di autenticazione per controllare la revoca del certificato TLS.
 - a) Definire un elenco nomi contenente gli oggetti delle informazioni di autenticazione.
 - b) Impostare l'attributo del gestore code **SSLCRLNL** sul nome dell'elenco nomi.

Concetti correlati

[Utilizzo dei certificati revocati](#)

Attività correlate

“Configurazione di un formato JSON CCDT” a pagina 46

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Si utilizza un editor di testo per creare e aggiornare un JSON (JavaScript Object Notation) CCDT.

Configurazione di un formato JSON CCDT

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Si utilizza un editor di testo per creare e aggiornare un JSON (JavaScript Object Notation) CCDT.

Prima di iniziare

Multi Se si utilizza IBM MQ for Multiplatforms, è possibile utilizzare invece la CCDT binaria che viene creata automaticamente quando si crea un gestore code. Consultare [“Configurazione di un formato binario CCDT”](#) a pagina 44.

Informazioni su questa attività

Il nome file dello schema CCDT per il formato JSON è:

Linux

`/opt/mqm/lib/ccdt_schema.json`

Windows

`C:\Program Files\IBM\MQ\bin\ccdt_schema.json`

Non esiste alcuna CCDT JSON predefinita e IBM MQ non fornisce strumenti per creare o modificare CCDT in formato JSON. Tuttavia, hai più opzioni di configurazione quando sviluppi manualmente una CCDT JSON rispetto a quando utilizzi il comando `runmqsc` per lavorare con una CCDT binaria:




- Non è necessario utilizzare IBM MQ for Multiplatforms per creare e modificare un file CCDT JSON.
- Utilizzando il formato JSON, è possibile definire definizioni di canale duplicate con lo stesso nome. Quando si distribuisce IBM MQ sul cloud, è possibile utilizzarlo per rendere la distribuzione scalabile e altamente disponibile.
- Il file JSON è leggibile, che può semplificare la configurazione del gestore code.
- Un formato di file flat può essere integrato con:
 - Strumenti di controllo della versione per tenere traccia della cronologia CCDT
 - Strumenti di automazione nella fornitura continua
- Non sono necessari strumenti specifici per gestire il file CCDT.

- Il file è più piccolo.
- Questo formato fornisce compatibilità con le versioni precedenti e successive.

Note:

1. Lo standard JSON vede le chiavi duplicate come valide, tuttavia, il programma di analisi JSON prende solo l'ultimo valore di lettura delle chiavi duplicate durante l'assegnazione degli attributi. Pertanto, quando si definiscono canali duplicati, ogni canale deve essere un elemento di un valore di array assegnato alla chiave 'channel'.
2. I CCDT JSON non supportano la memorizzazione delle ubicazioni del server LDAP (Lightweight Directory Access Protocol) per le informazioni sull'ubicazione del responder CRL (Certificate Revocation Lists) e OCSP (Online Certificate Status Protocol).

Tabella 7. Requisiti di codifica per piattaforma

Piattaforma	Codifica client JMS	Codifica client C
 IBM i	ASCII	EBCDIC
 AIX, Linux, and Windows	ASCII	ASCII
 z/OS	ASCII o EBCDIC	Non applicabile



Attenzione: Quando fornisci una definizione per un canale tramite una CCDT JSON (inclusa una *sparse* che non include tutti gli attributi), viene creata una definizione di canale completa con tutti gli attributi definiti, utilizzando i valori predefiniti per tutto ciò che non è specificato nel JSON.

Pertanto, è necessario fornire valori specifici per ogni attributo per cui non si desidera il valore predefinito.

Procedura

- Crea una CCDT JSON
 - a) Creare un file flat con estensione .json con un editor di testo generico.
 - b) Definire una CCDT.




Consultare [“Esempi CCDT JSON”](#) a pagina 50 e [“Attributi di canale supportati da JSON CCDT”](#) a pagina 48.
- Individuare la CCDT:
 - Sul computer client
 - In una posizione condivisa da più di un cliente
 - Sul server come file condiviso

Consultare [“Ubicazioni per la CCDT”](#) a pagina 53.
- Convalida una CCDT JSON

Convalida CCDT rispetto allo schema con un linter JSON.

Vedi [Come convalidare un file IBM MQ CCDT JSON file rispetto allo schema](#) per informazioni su come creare un file CCDT con due canali e convalidarne il funzionamento.

Lo schema CCDT è incluso con i package del prodotto e del client:

 -   Su sistemi AIX and Linux:
\$MQ_INSTALLATION_PATH/lib e /lib rispettivamente nei package del prodotto e client.
 -  Su Windows:
%MQ_INSTALLATION_PATH%\bin e \bin rispettivamente nei package del prodotto e client.

Note:

- I linters JSON sono disponibili online.
- Lo schema definisce gli attributi obbligatori con la chiave 'required'.
- Lo schema definisce i tipi di dati attributo con la chiave 'type'.
- Accedi alla CCDT
 - È possibile accedere alla CCDT:
 - In remoto da un file, ftp o URL http, definendo la variabile di ambiente **MQCCDTURL** .
 - Localmente impostando le variabili di ambiente **MQCHLLIB** e **MQCHLTAB** .
 - Localmente definendo gli attributi **ChannelDefinitionDirectory** e **ChannelDefinitionFile** della stanza CHANNELS nel file di configurazione del client.

Consultare [“Ubicazioni per la CCDT” a pagina 53](#) per vari esempi.

- Visualizzare o modificare il contenuto CCDT

Ogni voce di una CCDT rappresenta una connessione client a un gestore code specifico. È possibile visualizzare o modificare il contenuto CCDT con un editor di testo.

Se si desidera visualizzare solo la CCDT, è anche possibile eseguire questa operazione utilizzando il comando **runmqsc** come riportato di seguito:

1. Impostare le variabili di ambiente per fornire l'accesso alla CCDT, come descritto nel passo precedente.
2. Eseguire il comando `runmqsc -n` . Per ulteriori informazioni, consultare [runmqsc](#).
3. Eseguire il comando **DISPLAY CHANNEL**. Ad esempio, eseguire `DISPLAY CHANNEL(*)` .

Concetti correlati

[Utilizzo dei certificati revocati](#)

Attività correlate

[“Configurazione di un formato binario CCDT” a pagina 44](#)

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

[“Configurazione di un cluster uniforme” a pagina 425](#)

I cluster uniformi consentono alle applicazioni di essere progettate per la scalabilità e la disponibilità e possono connettersi a qualsiasi gestore code all'interno di tale cluster uniforme.

Attributi di canale supportati da JSON CCDT

Un elenco degli attributi del canale di connessione client supportati dalla CCDT JSON. Questo elenco è un sottoinsieme degli attributi supportati dal CCDT binario.

Associazione attributo

Questi attributi vengono inseriti nel seguente oggetto canale:

```
{ "channel": [ { $CHANNEL_1_KEY_VALUE_LIST }, ..., { $CHANNEL_N_KEY_VALUE_LIST } ] }
```

dove `$CHANNEL_X_KEY_VALUE_LIST` è un elenco separato da virgole degli attributi elencati nella seguente tabella.

Per i casi di utilizzo di base, consultare [“Esempi CCDT JSON” a pagina 50](#) .

Lo schema JSON viene fornito in `/opt/mqm/lib/ccdt_schema.json`. Per rilevare quali valori sono validi per ciascuno degli attributi, esaminare lo schema JSON.

La seguente tabella riporta l'oggetto JSON, la chiave e il tipo di dati, insieme alla corrispondente definizione di attributo del canale binario.



Attenzione: Gli attributi richiesti sono il canale **name** e il canale **type**. Se si definisce anche **portRange**, sono richiesti anche gli attributi *basso* e *alto*.

Oggetto JSON	Chiave JSON	Tipo di dati JSON	Definizione attributo binario
canale (array)	nome	Stringa	CHANNEL
canale (array)	il tipo	Stringa	CHLTYPE
channel.clientConnection	queueManager	Stringa	QMNAME
channel.clientConnection.connection (array)	host	Stringa	CONNNAME
channel.clientConnection.connection	porta	INT	CONNNAME
channel.compression.header (array)	intestazione	Stringa	COMPHDR
channel.compression.message (array)	messaggio	Stringa	COMPMSG
channel.connectionManagement	affinità	Stringa	AFFINITÀ
channel.connectionManagement	clientWeight	INT	CLNTWGHT
channel.connectionManagement	defaultReconnect	Stringa	DEFRECON
channel.connectionManagement	disconnectInterval	INT	DISCINT
channel.connectionManagement	heartInterval	INT	HBINT
channel.connectionManagement	KeepAliveInterval	INT	KAINT
channel.connectionManagement	sharingConversations	INT	SHARECNV
channel.connectionManagement.localAddress (array)	host	Stringa	LOCLADDR
channel.connectionManagement.localAddress (array)	porta	INT	LOCLADDR
channel.connectionManagement.localAddress.portRange	alto	INT	LOCLADDR
channel.connectionManagement.localAddress.portRange	basso	INT	LOCLADDR
channel.exits.receive (array)	nome	Stringa	RCVEXIT
channel.exits.receive (array)	userData	Stringa	RCVDATA
channel.exits.security	nome	Stringa	SCYEXIT
channel.exits.security	userData	Stringa	SCYDATA
channel.exits.send (array)	nome	Stringa	SENDEXIT
channel.exits.send (array)	userData	Stringa	SENDDATA
channel.general	descrizione	Stringa	DESCR
channel.general	maximumMessageLun ghezza	INT	MAXMSGSL
channel.timestamps	modificato	Stringa	ALTDATE e ALTTIME
channel.transmissionSecurity	certificateLabel	Stringa	CERTLABL
channel.transmissionSecurity	Nome certificatePeer	Stringa	SSLPEER

Oggetto JSON	Chiave JSON	Tipo di dati JSON	Definizione attributo binario
channel.transmissionSecurity	cipherSpecification	Stringa	SSLCIPH

Note:

- channel.connectionManagement.localAddress può essere definito come una delle seguenti combinazioni di chiavi:
 - Host e porta
 - host e portRange
 - porta
 - portRange
- La chiave JSON channel.timestamps.altered è facoltativa e, se non è definita, il valore predefinito è l'ora dell'ultima modifica del file CCDT JSON. Tuttavia, se l'ambiente è configurato per recuperare il CCDT da un URL, il valore predefinito è l'ora in cui il file è stato scaricato l'ultima volta.
- channel.clientConnection.connection deve includere entrambe le chiavi host e porta.
- La chiave modificata è una singola stringa che incapsula entrambi gli attributi ALTDATE e ALTTIME.
- Il tipo di trasporto può essere solo TCP, quindi i seguenti attributi non sono definiti nello schema:
 - **TRPTYPE**
 - **USERID**
 - **PASSWORD**
 - **MODENAME**
 - **TPNAME**

Riferimenti correlati

[Attributi del canale per i tipi di canale](#)

Esempi CCDT JSON

Utilizzare gli esempi elencati in questo argomento come base per i propri requisiti.

Aprire un editor di testo generico e copiare uno dei seguenti esempi:

- [“Definire una connessione client semplice” a pagina 50](#)
- [“Definire un canale e un gestore code utilizzando TLS” a pagina 51](#)
- [“Definire un canale e un gestore code che non utilizzano TLS” a pagina 51](#)
- [“Definire due canali con lo stesso nome” a pagina 51](#)
- [“Elenco completo di definizioni di attributi di canale CCDT per un canale di connessione client” a pagina 52](#)

Definire una connessione client semplice

```

{
  "channel":
  [
    {
      "general":
      {
        "description": "a channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [

```

```

    {
      "host": "localhost",
      "port": 1414
    }
  ],
  "queueManager": "QM1"
},
"type": "clientConnection"
}
]
}

```

Definire un canale e un gestore code utilizzando TLS

```

{
  "channel": [
    {
      "name": "SSL.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1419
          }
        ],
        "queueManager": "QM92TLS"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "TLS_AES_128_GCM_SHA256",
        "certificateLabel": "ibmwebspheremqadministrator",
      },
      "type": "clientConnection"
    }
  ]
}

```

Definire un canale e un gestore code che non utilizzano TLS

```

{
  "channel": [
    {
      "name": "SYSTEM.DEF.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1414
          }
        ],
        "queueManager": "QM92"
      },
      "type": "clientConnection"
    }
  ]
}

```

Definire due canali con lo stesso nome

Ogni canale si connette a due gestori code distinti:

```

{
  "channel":
  [
    {
      "general":
      {
        "description": "First channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":

```

```

    [
      {
        "host": "localhost",
        "port": 1414
      }
    ],
    "queueManager": "QM1"
  },
  "type": "clientConnection"
},
{
  "general":
  {
    "description": "Second channel"
  },
  "name": "channel",
  "clientConnection":
  {
    "connection":
    [
      {
        "host": "localhost",
        "port": 1415
      }
    ],
    "queueManager": "QM2"
  },
  "type": "clientConnection"
}
]
}

```

Elenco completo di definizioni di attributi di canale CCDT per un canale di connessione client

```

{
  "channel":
  [
    {
      "compression":
      {
        "header": [ "system" ],
        "message": [ "zlibfast" ]
      },
      "connectionManagement":
      {
        "sharingConversations": 10,
        "clientWeight": 1,
        "affinity": "none",
        "defaultReconnect": "yes",
        "heartbeatInterval": 600,
        "keepAliveInterval": -1,
        "localAddress":
        [
          {
            "portRange":
            {
              "low": 2020,
              "high": 3030
            }
          }
        ]
      }
    },
    {
      "exits":
      {
        "receive":
        [
          {
            "name": "",
            "userData": ""
          }
        ],
        "security":
        {
          "name": "",
          "userData": ""
        }
      },
      "send":

```

```

[
  {
    "name": "",
    "userData": ""
  }
],
"general":
{
  "description": "First channel",
  "maximumMessageLength": 4194304
},
"name": "the_channel",
"clientConnection":
{
  "connection":
  [
    {
      "host": "localhost",
      "port": 1414
    }
  ],
  "queueManager": "QM1"
},
"timestamps":
{
  "altered": "2018-12-04T15:37:22.000Z"
},
"transmissionSecurity":
{
  "cipherSpecification": "",
  "certificateLabel": "",
  "certificatePeerName": ""
},
"type": "clientConnection"
}
]
}

```

Riferimenti correlati

Attributi del canale per i tipi di canale

[Attributi del canale in ordine alfabetico](#)

Ubicazioni per la CCDT

IBM MQ supporta il richiamo di una CCDT da un file, FTP o URL HTTP. È possibile rendere il CCDT accessibile per il client come un file condiviso, mentre rimane sul server. In alternativa è possibile distribuire la CCDT, copiando la CCDT su singoli computer client o copiando la CCDT in una posizione condivisa da più di un client.

Se si utilizza FTP per copiare il file, utilizzare l'opzione `bin` per impostare la modalità binaria; non utilizzare la modalità ASCII predefinita. Qualunque sia il metodo scelto per rendere disponibile la CCDT, la posizione deve essere sicura per impedire modifiche non autorizzate ai canali.

Come ospitare il file CCDT su un server

La CCDT può essere ospitata in un'ubicazione centrale accessibile tramite un URL, eliminando la necessità di aggiornare singolarmente la CCDT per ogni client distribuito. Le applicazioni .NET native (C/C ++, COBOL e RPG) e non gestite possono estrarre la CCDT da un URL, sia che si tratti di un file locale, FTP o una risorsa HTTP.

Il comportamento di memorizzazione nella cache predefinito dei client IBM MQ è che un file CCDT viene estratto solo se l'ora di modifica del file è diversa dall'ultima volta che è stato richiamato. Come con la maggior parte delle opzioni di configurazione del client, ci sono diversi modi in cui è possibile fornire l'ubicazione dell'URL:

- **`CCDTUr1Ptr`** e **`CCDTUr1offset`** tramite la struttura `MQCNO` trasmessa nella chiamata `MQI MQCONN`
- **`MQCCDTURL`** variabile di ambiente
- Attributo **`ChannelDefinitionDirectory`** nella stanza `Channels` di `mqclient.ini`

Sono supportati sia URL autenticati che non autenticati. Di seguito sono riportati alcuni esempi:

```
export MQCCDTURL=ftp://myuser:password@myhost.sample.com//var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

```
export MQCCDTURL=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

Se si desidera utilizzare questo supporto con FTP o HTTP, è ancora necessario ospitare il file CCDT su un server, ma tutte le applicazioni client possono automaticamente rilevare le modifiche alle definizioni di canale senza eseguire manualmente il push degli aggiornamenti o la necessità di montare un filesystem di rete su ciascun client. Per ulteriori informazioni, consultare [“Accesso URL alla CCDT” a pagina 54](#).

Come specificare l'ubicazione della CCDT sul client

Su un sistema client, è possibile specificare l'ubicazione della CCDT nei seguenti modi:

- Utilizzo delle variabili di ambiente **MQCHLLIB** per specificare la directory in cui si trova la tabella e **MQCHLTAB** per specificare il nome file della tabella.
- Utilizzo del file di configurazione client. Nella stanza CHANNELS, utilizzare l'attributo **ChannelDefinitionDirectory** per specificare la directory in cui si trova la tabella e l'attributo **ChannelDefinitionFile** per specificare il nome file.
- Fornendo un URL (file, FTP o HTTP) per una CCDT che si trova in una posizione centrale come descritto in precedenza.

Se l'ubicazione è specificata sia nel file di configurazione client che utilizzando le variabili di ambiente, le variabili di ambiente hanno la priorità. È possibile utilizzare questa funzione per specificare un'ubicazione standard nel file di configurazione del client e sovrascriverla utilizzando le variabili di ambiente quando necessario.

Se si utilizza un URL per fornire l'ubicazione della CCDT, l'ordine di precedenza per un'applicazione del client nativo per trovare la definizione del canale client è come descritto in [“Accesso URL alla CCDT” a pagina 54](#).

Accesso URL alla CCDT

È possibile ospitare una CCDT (client channel definition table) in un'ubicazione centrale a cui è possibile accedere tramite un URL, eliminando la necessità di aggiornare singolarmente la CCDT per ciascun client distribuito.

Una tabella di definizione di canale client può essere individuata tramite un URL in uno dei seguenti modi:

- Programmando utilizzando MQCNO
- Utilizzando le variabili di ambiente



Attenzione: È possibile utilizzare l'opzione della variabile di ambiente per fornire l'URL solo per i programmi nativi che si collegano come client, ovvero le applicazioni C, COBOL o C++. Le variabili di ambiente non hanno effetto per le applicazioni Java, JMS o .NET gestite.

IBM MQ supporta il richiamo di una CCDT da un file, ftp o URL http.

- Utilizzando la stanza Canali del file `mqclient.ini`.

La variabile di ambiente **MQCCDTURL** consente di fornire un URL di file, ftp o http come un singolo valore da cui è possibile ottenere una tabella di definizione del canale client.

È anche possibile utilizzare il percorso di directory specificato dalla variabile di ambiente **MQCHLLIB** (o il percorso specificato dall'attributo **ChannelDefinitionDirectory** in “Stanza dei canali del file di configurazione client” a pagina 186) per individuare un file CCDT, tramite file, ftp o URL http, in aggiunta alla directory del file system locale esistente, ovvero `/var/mqm`). Si noti che un valore **MQCHLLIB** è una radice della directory e funziona in combinazione con **MQCHLTAB** per derivare l'URL completo.

L'autenticazione di base sulle connessioni è supportata tramite le credenziali codificate nell'URL:

Connessioni autenticate

```
export MQCHLLIB=ftp://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
```

Connessioni non autenticate

```
export MQCHLLIB=ftp://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=file:///var/mqm/qmgrs/QMGR/@ipcc
```

Nota: Se si desidera utilizzare connessioni autenticate, è necessario, come con JMS, fornire il nome utente e la password codificati nell'URL.

L'ordine di precedenza, per un'applicazione client nativa, per trovare una definizione di canale client è ora:

1. MQCD fornito da **ClientConnOffset** e **ClientConnPtr** in MQCNO.
2. URL fornito da **CCDTURLOffset** e **CCDTURLPtr** in MQCNO.
3. **MQSERVER** variabile di ambiente.
4. Se è definito un file `mqclient.ini` e la stanza Canali contiene un attributo **ServerConnectionParms**, viene utilizzato il canale che definisce. Per ulteriori informazioni, consultare [“IBM MQ MQI client file di configurazione, mqclient.ini”](#) a pagina 168 e [“Stanza dei canali del file di configurazione client”](#) a pagina 186.
5. **MQCCDTURL** variabile di ambiente.
6. **MQCHLLIB** e **MQCHLTAB** variabile di ambiente.
7. **ChannelDefinitionDirectory** e **ChannelDefinitionFile** in [“Stanza dei canali del file di configurazione client”](#) a pagina 186.

Importante: L'accesso a un file CCDT mediante un URL apre sempre una copia di sola lettura del file, anche quando viene utilizzato il protocollo `file://`.

Il tentativo di aprire un file CCDT per l'accesso in scrittura, ad esempio quando si utilizza il comando MQSC **DEFINE CHANNEL** da un client, restituisce un messaggio di errore che indica che non è stato possibile aprire il file per l'accesso in scrittura.

Tuttavia, è possibile leggere le definizioni delle informazioni di autenticazione e del canale utilizzando **runmqsc**.

Attività correlate

[“Accesso alle definizioni del canale di connessione client”](#) a pagina 58

È possibile rendere la tabella di definizione del canale client (CCDT) disponibile per le applicazioni client copiandola o condividendolo, quindi specificarne l'ubicazione e il nome sul computer client. È anche possibile individuare una tabella di definizione del canale client (CCDT) tramite un URL.

[“Configurazione di un formato binario CCDT”](#) a pagina 44

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

Utilizzo di CCDT con IBM MQ classes for JMS

Riferimenti correlati

[CCDTURL](#)

[MQCNO - Opzioni di connessione](#)

[URL CCDT WMQ_XMSC](#)



Canali di connessione client in Active Directory

Sui sistemi Windows che supportano Active Directory, IBM MQ pubblica i canali di connessione client in Active Directory per fornire il binding client-server dinamico.

Quando gli oggetti del canale di connessione client sono definiti, vengono scritti in un file di definizione del canale client, denominato AMQCLCHL . TAB per impostazione predefinita. Se i canali di connessione client utilizzano il protocollo TCP/IP, il server IBM MQ li pubblica anche in Active Directory. Quando il client IBM MQ stabilisce come connettersi al server, ricerca una definizione di oggetto del canale di connessione client pertinente utilizzando il seguente ordine di ricerca:

1. Struttura dati MQCD MQCONNX
2. **MQSERVER** variabile di ambiente
3. File di definizione del canale client
4. Active Directory

Questo ordine indica che le applicazioni correnti non sono interessate da alcuna modifica. È possibile considerare queste voci in Active Directory come record nel file di definizione del canale client e il client IBM MQ le elabora nello stesso modo. Per configurare e gestire il supporto per la pubblicazione di definizioni di canali di connessione client in Active Directory, utilizzare il comando `setmqscp`, come descritto in [setmqscp](#).

Definizione del canale di connessione server sul server

Creare una definizione di canale di connessione server per il gestore code.

Procedura

1. Sulla macchina server, definire un canale con il nome scelto e un tipo di canale *server - connection*. Ad esempio:

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_2')
```

2. Utilizzare il seguente comando per consentire l'accesso di connessione in entrata al gestore code:

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- Dove **SET CHLAUTH** utilizza il nome del canale definito nel passo precedente.
- Dove *'indirizzo IP'* l'indirizzo IP è l'indirizzo IP del client.
- Dove *'userid'* è l'ID che si desidera fornire al canale per il controllo dell'accesso alle code di destinazione. Questo campo è sensibile al maiuscolo / minuscolo.

È possibile scegliere di identificare la connessione in entrata utilizzando un numero di attributi differenti. L'esempio utilizza l'indirizzo IP. Gli attributi alternativi includono l'ID utente client e il DN (Distinguished Name) oggetto TLS. Per ulteriori informazioni, consultare [Record di autenticazione di canale](#)

Questa definizione di canale è associata al gestore code in esecuzione sul server.

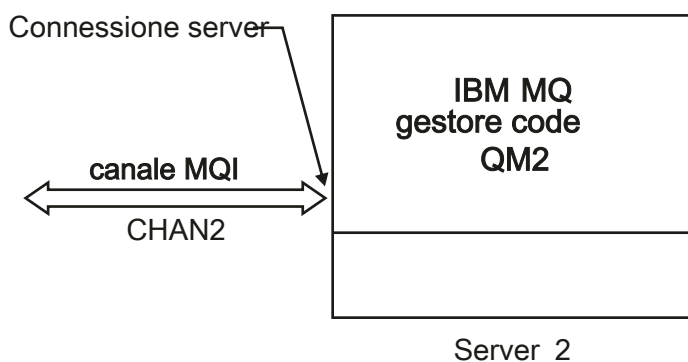


Figura 2. Definizione del canale di connessione server

Attività correlate

[“Definizione del canale di connessione client nel server” a pagina 57](#)

Una volta definito il canale di connessione server, è possibile definire il corrispondente canale di connessione client.

Definizione del canale di connessione client nel server

Una volta definito il canale di connessione server, è possibile definire il corrispondente canale di connessione client.

Prima di iniziare

Definire il canale di connessione server. Per ulteriori informazioni, consultare [“Definizione del canale di connessione server sul server” a pagina 56](#).

Procedura

1. Definire un canale con lo stesso nome del canale di connessione server, ma con un tipo di canale *connessione client*. È necessario indicare il nome connessione (CONNAME). Per TCP/IP, il nome della connessione è l'indirizzo di rete o il nome host della macchina server. È inoltre consigliabile specificare il nome del gestore code (QMNAME) a cui si desidera connettere l'applicazione IBM MQ, in esecuzione nell'ambiente client. Variando il nome del gestore code, è possibile definire una serie di canali per connettersi a gestori code differenti.

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```


2. Utilizzare il seguente comando per consentire l'accesso di connessione in entrata al gestore code:

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP-address') MCAUSER('userid')
```

- Dove il comando **SET CHLAUTH** utilizza il nome del canale definito nel passo precedente.
- Dove *indirizzo IP* è l'indirizzo IP del client.
- Dove *userid* è l'ID che si desidera fornire al canale per il controllo dell'accesso alle code di destinazione. Questo campo è sensibile al maiuscolo / minuscolo.

È possibile scegliere di identificare la connessione in entrata utilizzando un numero di attributi differenti. L'esempio utilizza l'indirizzo IP. Gli attributi alternativi includono l'ID utente client e il DN (Distinguished Name) oggetto TLS. Per ulteriori informazioni, consultare [Record di autenticazione di canale](#)

Risultati

 Su [Multiplatforme](#), questa definizione di canale viene memorizzata in un file denominato CCDT (client channel definition table), associato con il gestore code. La tabella di definizione del canale client può contenere più di una definizione del canale di connessione client.

Per ulteriori informazioni sulla tabella di definizione di canale client, e per le informazioni corrispondenti su come le definizioni di canale di connessione client sono memorizzate su z/OS, consultare [“Configurazione di un formato binario CCDT” a pagina 44](#).

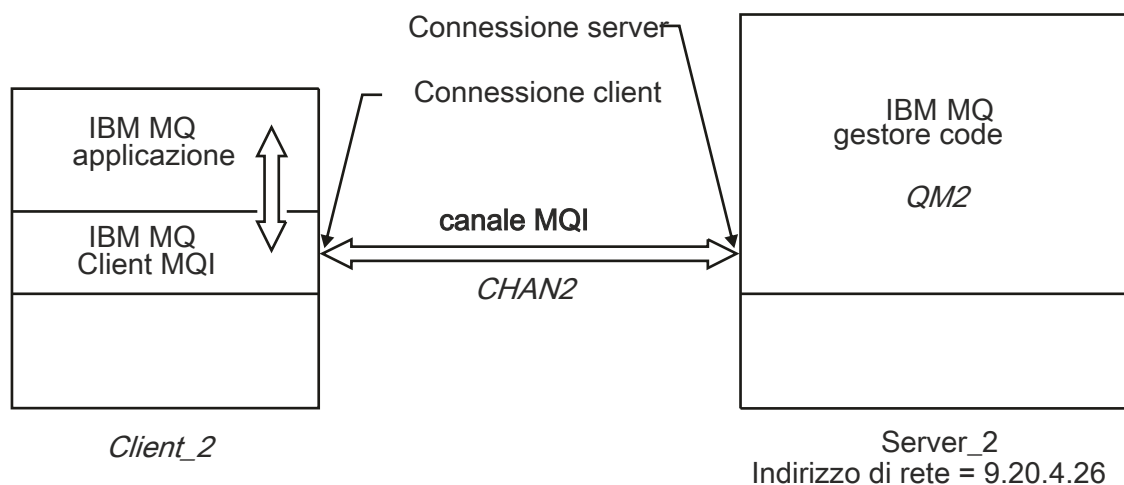


Figura 3. Definizione del canale di connessione client

Riferimenti correlati

[DEFINE CHANNEL](#) (definire un nuovo canale)

[SET CHLAUTH](#) (creazione o modifica di un record di autenticazione di canale)

Accesso alle definizioni del canale di connessione client

È possibile rendere la tabella di definizione del canale client (CCDT) disponibile per le applicazioni client copiandola o condividendola, quindi specificarne l'ubicazione e il nome sul computer client. È anche possibile individuare una tabella di definizione del canale client (CCDT) tramite un URL.

Prima di iniziare

Questa attività presuppone che siano stati definiti, in una CCDT, i canali di connessione client necessari. Consultare [“Configurazione delle tabelle di definizione del canale client”](#) a pagina 43.

Informazioni su questa attività

Affinché un'applicazione client utilizzi la CCDT (client channel definition table), è necessario renderla disponibile e specificarne l'ubicazione e il nome. Esistono diversi modi per eseguire questa operazione:

- È possibile copiare CCDT sul computer client.
- È possibile copiare la CCDT in un'ubicazione condivisa da più di un client.
- È possibile rendere il CCDT accessibile per il client come un file condiviso, mentre rimane sul server.

Le applicazioni IBM MQ native (C/C ++, COBOL e RPG) e .NET non gestite possono estrarre la CCDT ospitata in un'ubicazione centrale da un URL, che si tratti di un file locale, ftp o risorsa http.

Procedura

1. Rendere CCDT disponibile per le applicazioni client in uno dei seguenti modi:
 - a) Opzionale: Copiare la CCDT sul computer client.
 - b) Opzionale: Copiare la CCDT in una posizione condivisa da più di un client.
 - c) Opzionale: Lasciare CCDT sul server ma renderlo condivisibile dal client.
 - d) Opzionale: Definire un file locale, un URL ftp o http per un CCDT ospitato in un'ubicazione centrale in modo che le applicazioni native (C/C ++, COBOL e RPG) e .NET non gestite possano estrarre il CCDT da questo URL.

Qualunque sia la posizione scelta per la CCDT, la posizione deve essere sicura per evitare modifiche non autorizzate ai canali.

2. Sul client, specificare l'ubicazione e il nome del file che contiene CCDT in uno di tre modi:
 - a) Opzionale: Utilizzare la sezione CHANNELS del file di configurazione client. Per ulteriori informazioni, consultare [“Stanza dei canali del file di configurazione client”](#) a pagina 186.
 - b) Opzionale: Utilizzare le variabili di ambiente **MQCHLLIB** e **MQCHLTAB**.

Ad esempio, è possibile impostare le variabili di ambiente immettendo:

-  Su sistemi AIX and Linux:

```
export MQCHLLIB= MQ_INSTALLATION_PATH/qmgrs/ QUEUEMANAGERNAME /@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

-  Su sistemi IBM i:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE('/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/@ipcc')
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(AMQCLCHL.TAB)
```

dove *MQ_INSTALLATION_PATH* rappresenta la directory di alto livello in cui è installato IBM MQ .

- c) Opzionale: Solo su Windows , utilizzare il comando di controllo **setmqscp** per pubblicare le definizioni di canale di connessione client in Active Directory.
- d) Fornire l'ubicazione di una CCDT ospitata centralmente tramite un URL, programmando utilizzando MQCNO, utilizzando le variabili di ambiente o utilizzando le stanze del file `mqClient.ini` . Per ulteriori informazioni, consultare [“Ubicazioni per la CCDT”](#) a pagina 53 e [“Accesso URL alla CCDT”](#) a pagina 54.

Se la variabile di ambiente **MQSERVER** è impostata, un client IBM MQ utilizza la definizione di canale di connessione client specificata da **MQSERVER** come preferenza rispetto a qualsiasi definizione nella tabella di definizione di canale client.

Attività correlate

[“Configurazione di un formato binario CCDT”](#) a pagina 44

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

Riferimenti correlati

[Client MQI: CCDT \(Client Channel Definition Table\)](#)

 **ALW**

Programmi di uscita canale per canali MQI

Sono disponibili tre tipi di uscita canale per l'ambiente IBM MQ MQI client su AIX, Linux, and Windows.

Essi sono:

- Uscita invio
- Uscita ricezione
- Uscita di sicurezza

Queste uscite sono disponibili sia sul client che sull'estremità server del canale. Le uscite non sono disponibili per l'applicazione se si utilizza la variabile di ambiente MQSERVER. Le uscite dei canali sono descritte in [Programmi di uscita dei canali di messaggistica](#).

Le uscite di invio e ricezione funzionano insieme. Esistono diversi modi possibili in cui è possibile utilizzarle:

- Suddivisione e riassettaggio di un messaggio

- Compressione e decompressione dei dati in un messaggio (questa funzionalità viene fornita come parte di IBM MQ, ma è possibile utilizzare una tecnica di compressione differente)
- Crittografia e decrittografia dei dati utente (questa funzionalità viene fornita come parte di IBM MQ, ma è possibile utilizzare una diversa tecnica di crittografia)
- Registrazione su giornale di ogni messaggio inviato e ricevuto

È possibile utilizzare l'uscita di sicurezza per garantire che il server e client IBM MQ vengano identificati correttamente e per controllare l'accesso.

Se le uscite di invio o ricezione sul lato connessione server dell'istanza del canale devono eseguire chiamate MQI sulla connessione a cui sono associate, utilizzano l'handle di connessione fornito nel campo MQCXP Hconn . È necessario essere consapevoli che le uscite di invio e ricezione della connessione client non possono effettuare chiamate MQI.

Concetti correlati

[“Uscite di sicurezza su una connessione client” a pagina 60](#)

È possibile utilizzare i programmi di uscita di sicurezza per verificare che il partner all'altra estremità di un canale sia autentico. Considerazioni speciali si applicano quando un'uscita di sicurezza viene applicata a una connessione client.

[Uscite utente, uscite API e servizi installabili IBM MQ](#)

Attività correlate

[Estensione delle funzioni del gestore code](#)

Riferimenti correlati

[“Percorso delle uscite” a pagina 60](#)

Un percorso predefinito per l'ubicazione delle uscite del canale è definito nel file di configurazione client. Le uscite canale vengono caricate quando un canale viene inizializzato.

[“Identificazione della chiamata API in un programma di uscita di invio o ricezione” a pagina 62](#)

Quando si utilizzano i canali MQI per i client, il byte 10 del buffer dell'agent identifica la chiamata API in uso quando viene richiamata un'uscita di invio o ricezione. Ciò è utile per identificare quali flussi di canale includono i dati utente e potrebbe richiedere l'elaborazione come la crittografia o la firma digitale.

Percorso delle uscite

Un percorso predefinito per l'ubicazione delle uscite del canale è definito nel file di configurazione client. Le uscite canale vengono caricate quando un canale viene inizializzato.

Su sistemi AIX, Linux, and Windows , un file di configurazione client viene aggiunto al sistema durante l'installazione di IBM MQ MQI client. In questo file è definito un percorso predefinito per l'ubicazione delle uscite del canale sul client, utilizzando la stanza:

```
ClientExitPath:
ExitsDefaultPath= string
ExitsDefaultPath64= string
```

dove *stringa* è un percorso file in un formato appropriato per la piattaforma

Quando un canale viene inizializzato, dopo una chiamata MQCONN o MQCONNX , viene ricercato il file di configurazione del client. La stanza ClientExitPath viene letta e vengono caricate tutte le uscite del canale specificate nella definizione del canale.

Uscite di sicurezza su una connessione client

È possibile utilizzare i programmi di uscita di sicurezza per verificare che il partner all'altra estremità di un canale sia autentico. Considerazioni speciali si applicano quando un'uscita di sicurezza viene applicata a una connessione client.

La [Figura 4 a pagina 62](#) illustra l'utilizzo delle uscite di sicurezza in una connessione client, utilizzando il gestore autorizzazioni oggetto IBM MQ per autenticare un utente.

Il campo `SecurityParmsPtr` o `SecurityParmsOffset` nella struttura `MQCNO` è impostato dal client e ci sono uscite di sicurezza ad entrambe le estremità del canale. Una volta terminato il normale scambio di messaggi di sicurezza e quando il canale è pronto per essere eseguito, la struttura `MQCSP` viene passata all'uscita di protezione client. L'uscita può accedere alla struttura `MQCSP` utilizzando il campo `SecurityParms` nella struttura `MQCXP`. Il tipo di exit è impostato su `MQXR_SEC_PARMS`. L'uscita di sicurezza può modificare le credenziali nella struttura `MQCSP` o lasciarle invariate.

I dati restituiti dall'uscita vengono quindi inviati all'estremità di connessione server del canale. La struttura `MQCSP` viene ricreata sull'estremità della connessione server del canale e viene passata all'exit di sicurezza della connessione server. L'uscita può accedere alla struttura `MQCSP` utilizzando il campo `SecurityParms` nella struttura `MQCXP`. L'uscita di sicurezza riceve ed elabora questi dati. Questa elaborazione è in genere per annullare qualsiasi modifica apportata alle credenziali nella struttura `MQCSP` dall'uscita client, che vengono quindi utilizzate per autorizzare la connessione del gestore code. Si fa riferimento alla struttura `MQCSP` risultante utilizzando `SecurityParmsPtr` nella struttura `MQCNO` sul sistema del gestore code.

L'indirizzo di memoria restituito con il campo `SecurityParms` della struttura `MQCXP` deve restare indirizzabile e non modificato fino a `MQXR_TERM`. Un'uscita non deve invalidare o liberare nuovamente la memoria sul sistema prima che l'uscita venga richiamata per `MQXR_TERM`.

Se il campo `SecurityParmsPtr` o `SecurityParmsOffset` nella struttura `MQCNO` è impostato e vi è un'uscita di sicurezza solo ad un'estremità del canale, l'uscita di sicurezza riceve ed elabora la struttura `MQCSP`. Le azioni come la crittografia non sono appropriate per una singola uscita utente, poiché non esiste alcuna uscita per eseguire l'azione complementare.

Se i campi `SecurityParmsPtr` e `SecurityParmsOffset` nella struttura `MQCNO` non sono impostati e c'è un'uscita di sicurezza in una o in entrambe le estremità del canale, vengono richiamate le uscite di sicurezza. Entrambe le uscite di sicurezza possono restituire la propria struttura `MQCSP` indirizzata dal campo `SecurityParmsPtr`. L'uscita di sicurezza non viene richiamata nuovamente fino a quando non viene terminata (ExitReason di `MQXR_TERM`). Il writer di uscita può liberare la memoria utilizzata per `MQCSP` in tale fase.

Quando un'istanza del canale di connessione server condivide più di una conversazione, il pattern di chiamate all'uscita di sicurezza è limitato alla seconda e alle successive conversazioni.

Per la prima conversazione, il pattern è lo stesso come se l'istanza del canale non stesse condividendo le conversazioni. Per le seconde e successive conversazioni, l'uscita di sicurezza non viene mai richiamata con `MQXR_INIT`, `MQXR_INIT_SEC` o `MQXR_SEC_MSG`. Viene richiamato con `MQXR_SEC_PARMS`.

In un'istanza del canale con conversazioni condivise, `MQXR_TERM` viene richiamato solo per l'ultima conversazione in esecuzione.

Ogni conversazione ha l'opportunità nel richiamo `MQXR_SEC_PARMS` dell'uscita per modificare `MQCD`; all'estremità della connessione server del canale questa funzione può essere utile per variare, ad esempio, i valori `MCAUserIdentifier` o `LongMCAUserIdPtr` prima che venga effettuata la connessione al gestore code.

Server-connection exit	Client-connection exit
	Invoked with MQXR_INIT Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

Figura 4. Scambio avviato dalla connessione client con accordo per la connessione client utilizzando i parametri di sicurezza

Nota: Le applicazioni di uscita di sicurezza create prima del rilascio di IBM WebSphere MQ 7.1 potrebbero richiedere l'aggiornamento. Per ulteriori informazioni, consultare [Channel security exit programs](#).

ALW Identificazione della chiamata API in un programma di uscita di invio o ricezione

Quando si utilizzano i canali MQI per i client, il byte 10 del buffer dell'agent identifica la chiamata API in uso quando viene richiamata un'uscita di invio o ricezione. Ciò è utile per identificare quali flussi di canale includono i dati utente e potrebbe richiedere l'elaborazione come la crittografia o la firma digitale.

La seguente tabella mostra i dati che vengono visualizzati in byte 10 del flusso del canale quando viene elaborata una chiamata API.

Nota: Questi non sono gli unici valori di questo byte. Esistono altri valori **riservati**.

chiamata API	Valore del byte 10 per la richiesta	Valore del byte 10 per la risposta
MQCONN "1" a pagina 63, "2" a pagina 63	X'81 '	'91'
MQDISC "1" a pagina 63	X'82 '	X' 92 '

Tabella 8. Identificazione delle chiamate API (Continua)

chiamata API	Valore del byte 10 per la richiesta	Valore del byte 10 per la risposta
MQOPEN "3" a pagina 63	X'83 '	X' 93 '
MQCLOSE	84 '	'94'
MQGET "4" a pagina 63	X'85 '	X' 95 '
MQPUT "4" a pagina 63	X'86 '	X' 96 '
MQPUT1 richiesta "4" a pagina 63	X'87 '	X' 97 '
Richiesta MQSET	X'88 '	'98'
Richiesta MQINQ	X'89 '	X' 99 '
Richiesta MQCMIT	X'8A'	X'9A'
Richiesta MQBACK	'8B'	X'9B'
Richiesta MQSTAT	'8D'	X'9D'
Richiesta MQSUB	'8E'	'9E'
Richiesta MQSUBRQ	'8F'	'9F'
richiesta xa_start	X'A1'	X'B1'
richiesta xa_end	X'A2'	X'B2'
Richiesta xa_open	X'A3'	X'B3'
richiesta xa_close	X'A4'	'B4'
Richiesta xa_prepare	'A5'	X'B5'
richiesta xa_commit	'A6'	'B6'
richiesta x_rollback	X'A7'	X'B7'
richiesta xa_forget	'A8'	X'B8'
richiesta xa_recover	X'A9'	'B9'
Richiesta xa_complete	X'AA'	X'BA '

Note:

1. La connessione tra il server e il client viene avviata dall'applicazione client utilizzando MQCONN. Pertanto, per questo comando in particolare, esistono diversi altri flussi di rete. Lo stesso vale per MQDISC, che termina la connessione di rete.
2. MQCONNX viene trattato allo stesso modo di MQCONN per la connessione client - server.
3. Se viene aperto un elenco di distribuzione di grandi dimensioni, è possibile che vi sia più di un flusso di rete per ogni chiamata MQOPEN per passare tutti i dati richiesti all'MCA SVRCONN.
4. I messaggi di grandi dimensioni possono superare la dimensione del segmento di trasmissione. Se ciò si verifica, possono essere presenti molti flussi di rete risultanti da una singola chiamata API.

z/OS Connecting a client to a queue sharing group

You can connect a client to a queue sharing group by creating an MQI channel between a client and a queue manager on a server that is a member of a queue sharing group.

About this task

A queue sharing group is formed by a set of queue managers that can access the same set of shared queues. For more information about shared queues, see [Shared queues and queue sharing groups](#).

A client putting to a shared queue can connect to any member of the queue sharing group. The benefits of connecting to a queue sharing group are possible increases in front-end and back-end availability, and increased capacity. You can connect to a specific queue manager or to the generic interface.

Connecting directly to a queue manager in a queue sharing group gives the benefit that you can put messages to a shared target queue, which increases back-end availability.

Connecting to the generic interface of a queue sharing group opens a session with one of the queue managers in the group. This increases front-end availability, because the client queue manager can connect with any queue manager in the group. You connect to the group using the generic interface when you do not want to connect to a specific queue manager within the queue sharing group.

The generic interface can be a Sysplex Distributor VIPA address or a VTAM generic resource name, or another common interface to the queue sharing group. For more details on setting up a generic interface, see [Setting up communication for IBM MQ for z/OS using queue sharing groups](#).

Procedure

To connect to the generic interface of a queue sharing group you need to create channel definitions that can be accessed by any queue manager in the group. To do this you must have the same definitions on each queue manager in the group.

1. Define the SVRCONN channel as shown in the following example:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

Channel definitions on the server are stored in a shared Db2® repository. Each queue manager in the queue sharing group makes a local copy of the definition, ensuring that you will always connect to the correct server-connection channel when you issue an MQCONN or MQCONNX call.

2. Define the CLNTCONN channel as shown in the following example:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME( VIPA address ) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

Results

Because the generic interface of the queue sharing group is stored in the CONNAME field in the client-connection channel, you can now connect to any queue manager in the group, and put to shared queues owned by that group.

Utilizzo delle variabili d'ambiente IBM MQ

È possibile utilizzare i comandi per visualizzare le impostazioni correnti o per reimpostare i valori delle variabili di ambiente IBM MQ.

Informazioni su questa attività

È possibile utilizzare le variabili di ambiente nei modi seguenti:

- Per impostare le variabili nel proprio profilo di sistema per effettuare una modifica permanente
- Per immettere un comando dalla riga comandi per effettuare una modifica solo per questa sessione
- Per assegnare a una o più variabili un determinato valore in base all'applicazione in esecuzione, aggiungere i comandi ad un file di script di comandi utilizzato dall'applicazione

Per ogni variabile di ambiente, è possibile utilizzare i comandi per visualizzare l'impostazione corrente o per reimpostare il valore della variabile di ambiente. Questi comandi sono disponibili su tutte le piattaforme supportate, se non diversamente specificato. Il formato del comando dipende dalla piattaforma. Ad esempio:

- **Linux** **AIX** Su AIX and Linux:

```
export [environment variable]=value
```

- **Windows** Su Windows:

```
Set [environment variable]=value
```

- **IBM i** Su IBM i:

```
ADDENVVAR ENVVAR(environment variable) VALUE(xx)
```

- **MQ Appliance** Per IBM MQ Appliance, vedi [Configuring environment variables on IBM MQ Appliance](#) nella documentazione IBM MQ Appliance .

Dove applicabile, IBM MQ utilizza i valori predefiniti per le variabili di ambiente che non sono state impostate.

Nota: **z/OS** IBM MQ for z/OS non supporta alcuna variabile di ambiente IBM MQ . Se si utilizza questa piattaforma come server, consultare [Tabella di definizione del canale client](#) per informazioni su come viene generata la tabella di definizione del canale client su z/OS. Puoi ancora usare le variabili di ambiente IBM MQ sulla tua piattaforma client.

Procedura

- **Windows**

Su Windows, per ogni variabile di ambiente, utilizzare i seguenti comandi per visualizzare l'impostazione corrente o per reimpostare il valore di una variabile:

- Per eliminare il valore di una variabile di ambiente, utilizzare il comando seguente:

```
SET MQSERVER=
```

- Per visualizzare l'impostazione corrente di una variabile di ambiente, utilizzare il comando seguente:

```
SET MQSERVER
```

- Per visualizzare tutte le variabili di ambiente per la sessione, utilizzare il seguente comando:

```
set
```

- **Linux** **AIX**

Su AIX and Linux, per ogni variabile di ambiente, utilizzare i seguenti comandi per visualizzare l'impostazione corrente o per reimpostare il valore di una variabile:

- Per eliminare il valore di una variabile di ambiente, utilizzare il comando seguente:

```
unset MQSERVER
```

- Per visualizzare l'impostazione corrente di una variabile di ambiente, utilizzare il comando seguente:

```
echo $MQSERVER
```

- Per visualizzare tutte le variabili di ambiente per la sessione, utilizzare il seguente comando:

```
set
```

Attività correlate

[Impostazione delle variabili di ambiente per IBM MQ classes for JMS/Jakarta Messaging](#)

[Variabili di ambiente relative a IBM MQ classes for Java](#)

[Definizione di variabili di ambiente aggiuntive nel file service.env](#)

“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms” a pagina 93

È possibile modificare il comportamento di IBM MQ o di un singolo gestore code per adattarlo alle esigenze della propria installazione modificando le informazioni nei file di configurazione (.ini). È anche possibile modificare le opzioni di configurazione per IBM MQ MQI clients.

Riferimenti correlati

[Utilizzo delle variabili di ambiente nelle proprietà MFT](#)

Descrizioni delle variabili di ambiente

Descrizioni delle variabili di ambiente server e client destinate all'utilizzo da parte del cliente.

Esempi di utilizzo

- **Linux** ► **AIX** Su sistemi AIX and Linux , utilizzare questo formato: export [environment variable]=value.
- **Windows** Su sistemi Windows , utilizzare questo formato: Set [environment variable]=value.
- **IBM i** Su sistemi IBM i , utilizzare questo formato: ADDENVVAR ENVVAR(environment variable) VALUE(xx).
- **MQ Appliance** Per IBM MQ Appliance, vedi [Configuring environment variables on IBM MQ Appliance](#) nella documentazione IBM MQ Appliance .

Tabella 9. Riepilogo delle variabili di ambiente

Nome	Descrizione	Sul server?	Sul client?
AMQ_ALLOWED_CIPHERS	Specifica un elenco personalizzato di CipherSpecs abilitati per l'utilizzo con i canali IBM MQ .	✓	
DATA_FDCS_BAD_AMQ	Fa sì che i file FFST vengano scritti per tutti i dati non corretti, inclusi i formati noti semplici, se IBM MQ riceve dati in un formato non corretto da un host su TCP/IP.	✓	
AMQ_CONVEBCDICNE_WLINE	Specifica come IBM MQ deve convertire un carattere EBCDIC NL in formato ASCII.	✓	
AMQ_DIAGNOSTIC_MSG_SEVERITY	Specifica se la severità del messaggio deve essere accodata al numero del messaggio quando un processo IBM MQ scrive un messaggio in un log degli errori o nella console.	✓	
AMQ_DISABLE_CLIENT_AMS	Per i client Java , se si desidera connettersi a un gestore code da una versione precedente del prodotto, disabilita IBM MQ Advanced Message Security (AMS) sul client.		✓

Tabella 9. Riepilogo delle variabili di ambiente (Continua)

Nome	Descrizione	Sul server?	Sul client?
<u>AMQ_DMPMQCFG_QS</u> <u>GDISP_DEFAULT</u>	Specifica quali tipi di definizioni includere quando si interroga la disposizione di un gestore code.	✓	
V9.4.0 <u>AMQ_IODELAY</u> , <u>AMQ_IODELAY_INMS</u> e <u>AMQ_IODELAY_FFST</u>	Utilizzato per ottimizzare la diagnostica e le tempistiche quando l'input e l'output per il file system di archiviazione e il log del gestore code sono lenti.	✓	
<u>TRACCIA_LDAP_AMQ</u>	Consente di attivare e disattivare la traccia del client LDAP senza arrestare o avviare anche il gestore code.	✓	
<u>AMQ_LICENSING_MER</u> <u>TRICA</u>	Fa in modo che il gestore code carichi i dati relativi a tipi di licenze VPC mensili invece del comportamento predefinito di caricamento dei dati correlati alle licenze orarie basate sul contenitore.	✓	
Linux AIX <u>AMQ_MQS_INI_LOCA</u> <u>TION</u>	Specifica l'ubicazione utilizzata per il file <code>mqs.ini</code> .	✓	
<u>NO_AMQ_BAD_COMM</u> <u>S_DATA_FDCS</u>	Elimina la generazione di FFST quando si riportano messaggi di errore sul flusso di comunicazioni iniziale se i dati che IBM MQ riceve da un host su TCP/IP hanno un formato non corretto.	✓	
<u>AMQ_NO_IPV6</u>	Disabilita l'utilizzo di IPv6 durante il tentativo di connessione.	✓	✓
<u>AMQ_REVERSE_COMM</u> <u>IT_ORDER</u>	Configura un gestore code in modo che in una transazione XA venga eseguito il commit della modifica del gestore code IBM MQ una volta completato l'aggiornamento del database corrispondente.	✓	
<u>AMQ_SSL_ALLOW_DE</u> <u>FAULT_CERT</u>	Consente al certificato che un'applicazione utilizza per connettersi a un gestore code di essere un certificato predefinito, purché sia presente un certificato predefinito nel repository delle chiavi e il repository delle chiavi non contenga un certificato personale con prefisso <code>ibmwebsphereuserid</code> .	✓	
<u>AMQ_SSL_LDAP_SERV</u> <u>ER_VERSION</u>	Specifica che LDAP v2 o LDAP v3 viene utilizzato dai componenti crittografici IBM MQ nei casi in cui i server CRL richiedono l'uso di una versione specifica del protocollo LDAP.	✓	

Tabella 9. Riepilogo delle variabili di ambiente (Continua)







Nome	Descrizione	Sul server?	Sul client?
 AMQ_USE_ZLIBNX	<p>Su AIX, abilita gli MCA (message channel agent) ad utilizzare la libreria zlibNX accelerata dall'hardware per la compressione e la decompressione dei dati dei messaggi quando si utilizzano tecniche ZLIBFAST o ZLIBHIGH.</p>	✓	
MQ_GMQ_LIB	<p>Specifica la libreria di bind del client quando le classi di automazione IBM MQ per ActiveX (MQAX) vengono eseguite sul client anziché sul server.</p> <p> Attenzione: Questa variabile di ambiente viene eliminata alla versione 9.2.</p>		
   CASA	<p>Su AIX, Linux e IBM i, specifica il nome della directory in cui viene ricercato il file <code>mqclient.ini</code>.</p>	✓	
 HOMEDRIVE e HOMEPATH	<p>Su Windows, utilizzato insieme per specificare il nome della directory in cui viene ricercato il file <code>mqclient.ini</code>.</p>	✓	
LDAP_BASEDN	<p>Ambiente richiesto per l'esecuzione di un programma di esempio LDAP. Specifica il DN (Distinguished Name) di base per la ricerca nell'indirizzario.</p>	✓	
HOST_LDAP	<p>Facoltativo per l'esecuzione di un programma di esempio LDAP. Specifica il nome dell'host su cui è in esecuzione il server LDAP.</p>	✓	
VERSIONE LDAP	<p>Facoltativo per l'esecuzione di un programma di esempio LDAP. Specifica la versione del protocollo LDAP da utilizzare.</p>	✓	
MQ_CHANNEL_SUPPRESS_INTERVAL	<p>Specifica l'intervallo di tempo, in secondi, durante il quale i messaggi definiti con MQ_CHANNEL_SUPPRESS_MSGS devono essere eliminati dalla scrittura nel log degli errori, insieme al numero di volte in cui un messaggio può verificarsi durante l'intervallo di tempo specificato prima di essere eliminato.</p>	✓	
MQ_CHANNEL_SUPPRESS_MSGS	<p>Elimina i messaggi di errore del canale nel log degli errori.</p>	✓	
TIPO_CONNET_MQ	<p>Utilizzato in combinazione con il tipo di bind specificato nel campo Opzioni della struttura MQCNO utilizzato su una chiamata MQCONN.</p>		✓

Tabella 9. Riepilogo delle variabili di ambiente (Continua)

Nome	Descrizione	Sul server?	Sul client?
<u>MQ_CROSS_QUEUE_ORDER_ALL</u>	Specifica che l'ordine di inserimento del messaggio viene conservato in un'unità di lavoro.	✓	
<u>MQ_EPHEMER_PREFIX</u>	Specifica il percorso della directory temporanea del gestore code, all'interno della quale vengono conservati i dati del gestore code temporaneo, mentre il gestore code è in esecuzione.	✓	
Windows <u>MQ_FILE_PATH</u>	Creato durante l'installazione del package di runtime. Contiene gli stessi dati del registro Windows .	✓	
<u>MQ_JAVA_DATA_PATH</u>	Specifica la directory per l'output di log e traccia per IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaginge IBM MQ classes for Java.	✓	
<u>PERCORSO_INSTALL_JAVA_MQ</u>	Specifica la directory in cui sono installati IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaginge IBM MQ classes for Java .	✓	
<u>MQ_JAVA_LIB_PATH</u>	Specifica la directory in cui sono memorizzate le librerie IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaginge IBM MQ classes for Java .	✓	
<u>"MQ_OVERRIDE_DATA_PATH"</u> a pagina 79	Sovrascrive la directory predefinita del percorso dati IBM MQ .	✓	✓
AIX <u>MQ_SET_NODELAYACK</u>	Su AIX, disattiva il riconoscimento ritardato TCP.	✓	
<u>"MQ_XX_ENCODE_CASE_ONE nome_utente"</u> a pagina 79	Consente a un'installazione non registrata su Linux di selezionare il nome di un utente senza nome.	✓	
<u>FILE_LOG_MQAPI</u>	Definisce il prefisso per il file specificato dall'utente in cui il programma di uscita API di esempio genera una traccia MQI.	✓	
ALW <u>MQAPPLNAME</u>	Se il nome dell'applicazione non è stato ancora scelto, specifica il nome da utilizzare per identificare la connessione a un gestore code.		✓
<u>MQCCSID</u>	Specifica il numero della serie di caratteri codificati da utilizzare e sostituisce il valore CCSID con cui è stato configurato il server.		✓

Tabella 9. Riepilogo delle variabili di ambiente (Continua)



Nome	Descrizione	Sul server?	Sul client?
URL MQCC	Fornisce la capacità equivalente di impostare una combinazione delle variabili di ambiente MQCHLLIB e MQCHLTAB .		✓
MQCERTLABL	Definisce l'etichetta del certificato di una definizione di canale che IBM MQ deve utilizzare per individuare un certificato personale inviato durante un handshake TLS.		✓
MQCERTVPOL	Specifica il tipo di politica di convalida del certificato da utilizzare.		✓
MQCHLLIB	Specifica il percorso della directory del file che contiene la tabella di definizione del canale client (CCDT).		✓
MQCHLTAB	Specifica il nome del file che contiene la CCDT (client channel definition table).		✓
MQCLNTCF	Specifica l'ubicazione del file di configurazione IBM MQ MQI client.		✓
TRACE_MQDOTNET_ON	Abilita la traccia per i client ridistribuibili IBM MQ .NET.		✓
MQIPADDRV	Specifica quale protocollo IP utilizzare per una connessione di canale.		✓
MQKEYRPWD	Specifica la password per il repository chiavi che contiene il certificato digitale appartenente all'utente.		✓
 MQLICENSE	Su sistemi Linux, utilizzato per accettare o visualizzare una licenza IBM MQ dopo l'installazione del prodotto.	✓	✓
MQMAXERRORLOGSIZE	Specifica la dimensione del log degli errori del gestore code copiato nel backup.	✓	
 NOME	Specifica il nome NetBIOS locale che i processi IBM MQ possono utilizzare.	✓	✓
MQNOREMPOOL	Disattiva il pool di canali e causa l'esecuzione dei canali come thread del listener.	✓	
LOGFILE MQPSE_TRACE_	Descrive dove devono essere scritti i file di traccia per il programma di esempio di uscita di pubblicazione.	✓	
MQS_AMSCRED_XX_ENCODE_CASE_ONE file_chiave	Sovrascrive o fornisce il file di chiavi iniziale da utilizzare al runtime delle applicazioni IBM MQ Advanced Message Security (AMS) o quando si sta proteggendo un file di configurazione del keystore utilizzando il comando runamscred .		✓

Tabella 9. Riepilogo delle variabili di ambiente (Continua)


Nome	Descrizione	Sul server?	Sul client?
MQS_DISABLE_ALL_IN TERCEPT	Per i client C nativi, se si desidera connettersi a un gestore code da una versione precedente del prodotto, disabilita IBM MQ Advanced Message Security (AMS) sul client.		✓
HOST_IP_MQS	Imposta il nome host che viene aggiunto al percorso della directory.	✓	
MQS_KEystore_CONF	Specifica l'ubicazione del file di configurazione del keystore per IBM MQ Advanced Message Security (AMS), se il file non si trova nell'ubicazione predefinita.		✓
MQS_MQI_XX_ENCODE_CASE_ONE file_chiave	Specifica l'ubicazione di un file di chiavi iniziale che contiene la chiave iniziale da utilizzare per le operazioni di protezione della parola d'ordine.		✓
MQS_SSLCRYP_XX_ENCODE_CASE_ONE file_chiave	Specifica il nome e il percorso completo del file contenente la chiave iniziale utilizzata per codificare la password nella stringa di configurazione hardware crittografica PKCS #11 .		✓
MQS_TRACE_OPZIONI	Per la traccia selettiva del componente su AIX, attiva singolarmente le funzioni di traccia dei parametri e dei dettagli elevati.	✓	✓
SERVER_MQT	Definisce un canale minimo, specificando l'ubicazione del server IBM MQ e il metodo di comunicazione da utilizzare.		✓
MQSNOAUT	Disabilita OAM (object authority manager) e impedisce qualsiasi controllo di sicurezza, ad esempio in un ambiente di test.	✓	
MQSPREFIX	In alternativa alla modifica del prefisso predefinito, sovrascrive DefaultPrefix per il comando crtmqm .	✓	
 MQSSLCRYP	Contiene una stringa di parametri che è possibile utilizzare per configurare l'hardware crittografico presente nel sistema.		✓
MQSSLFIPS	Specifica se devono essere utilizzati solo algoritmi certificati FIPS se la crittografia viene eseguita in IBM MQ.		✓
MQSSLKEYR	Specifica l'ubicazione del repository chiavi che contiene il certificato digitale appartenente all'utente.		Strumento client e mqcertck ✓

Tabella 9. Riepilogo delle variabili di ambiente (Continua)

Nome	Descrizione	Sul server?	Sul client?
<u>MQSSLPROXY</u>	Specifica il numero di porta e il nome host del server proxy HTTP che deve essere utilizzato da IBM Global Security Kit (GSKit) per i controlli OCSP.		✓
<u>RESET MQSS</u>	Specifica il numero di byte non codificati inviati e ricevuti su un canale TLS prima che la chiave segreta TLS venga rinegoziata.		✓
ALW <u>MQSUITEB</u>	Specifica se deve essere utilizzata la crittografia conforme a Suite B. Nell'istanza in cui viene utilizzata la crittografia Suite B,		✓
<u>MQTCPTIMEOUT</u>	specifica per quanto tempo IBM MQ attende una chiamata di connessione TCP.	✓	✓
<u>ODQ_MSG</u>	Imposta il nome del file contenente i messaggi di errore e di informazione, se si utilizza un gestore code di messaggi non instradabili diverso da runmqdlq .	✓	
<u>TRACCIA_ODQ</u>	Abilita la traccia se si utilizza un gestore code di messaggi non recapitabili diverso da runmqdlq .	✓	
<u>TRACE_WCF</u>	Abilita la traccia per il canale personalizzato WCF.		✓
<u>WMQSOAP_HOME</u>	Utilizzato quando si completano ulteriori operazioni di configurazione dopo che l'ambiente di hosting del servizio .NET SOAP over JMS viene installato e configurato correttamente in IBM MQ.		✓
<u>XMS_TRACE_ON</u> , <u>XMS_TRACE_FILE_PATH</u> , <u>XMS_TRACE_FORMAT</u> e <u>XMS_TRACE_SPECIFICATION</u>	Utilizzato per abilitare e configurare la traccia XMS .		✓

AMQ_ALLOWED_CIPHERS

Multi

È possibile utilizzare la variabile di ambiente **AMQ_ALLOWED_CIPHERS** per specificare un elenco personalizzato di CipherSpecs abilitati per l'utilizzo con i canali IBM MQ su Multiplatforms. La variabile di ambiente assume gli stessi valori dell'attributo della stanza **AllowedCipherSpecs** SSL del file `.ini` :

- Un singolo nome CipherSpec oppure
- Un elenco separato da virgole di nomi IBM MQ CipherSpec da riabilitare o
- Il valore speciale di ALL, che rappresenta tutte le CipherSpecs (non consigliato).

Nota: L'abilitazione di **ALL** CipherSpecs non è consigliata poiché abiliterà i protocolli SSL 3.0 e TLS 1.0 e un numero elevato di algoritmi crittografici deboli.

Per ulteriori informazioni, vedi [Fornisci un elenco personalizzato di CipherSpecs abilitati su Multiplatforms nell'ordine CipherSpec nell'handshake TLS](#).

FDCS_DATA_BAD_AMQ_

La variabile di ambiente **AMQ_BAD_COMMS_DATA_FDCS** è valida quando è impostata su qualsiasi valore.

Se i dati che IBM MQ riceve da un host su TCP/IP sono in un formato non corretto, ad esempio perché un client di rete si è connesso a una porta del listener IBM MQ e ha tentato di comunicare con un protocollo di applicazione non supportato, il gestore code scrive un messaggio di errore **AMQ9207E** nei log di errori del gestore code. I listener IBM MQ supportano le connessioni TCP/IP da MCA (message channel agent) del gestore code e da applicazioni client MQI, JMS e XMS .

Nota: I listener IBM MQ non supportano il protocollo dell'applicazione utilizzato dai client AMQP e MQTT, questi client devono invece connettersi alle porte di rete configurate nel canale AMQP o nel servizio di telemetria MQXR applicabile.

Potrebbe essere scritto anche un record FDC (failure data capture) contenente i dati non validi ricevuti da IBM MQ . Tuttavia, un file FFST non viene generato se questo è l'inizio di una conversazione con il lato remoto e il formato è un semplice formato noto come una richiesta GET da un browser web HTTP. Se si desidera sovrascrivere questo valore per far sì che i file FFST vengano scritti per qualsiasi dato non valido, inclusi i semplici formati noti, è possibile impostare la variabile di ambiente **AMQ_BAD_COMMS_DATA_FDCS** su qualsiasi valore (ad esempio, TRUE) e riavviare il gestore code.

AMQ_TRASPORTBCDICNEWLINE



È possibile utilizzare la variabile di ambiente **AMQ_CONVEBCDICNEWLINE** per specificare come IBM MQ deve convertire un carattere EBCDIC NL in formato ASCII. La variabile di ambiente assume gli stessi valori dell'attributo **ConvEBCDICNewLine** di `mqs.ini`, ovvero, `NL_TO_LF`, `TABLE` o `ISO` (consultare [Stanza di tutti i gestori code del file mqs.ini](#)). È possibile, ad esempio, utilizzare la variabile di ambiente **AMQ_CONVEBCDICNEWLINE** invece dell'attributo della stanza **ConvEBCDICNewLine** per fornire la funzionalità **ConvEBCDICNewLine** sul lato client in situazioni in cui non è possibile utilizzare il file `mqs.ini` . Se sono impostati sia l'attributo stanza che la variabile di ambiente, l'attributo stanza ha la precedenza.

Per ulteriori informazioni, consultare [Conversione dei dati tra serie di caratteri codificati](#) .

AMQ_DIAGNOSTIC_MSG_SEVERITY

Se la variabile di ambiente **AMQ_DIAGNOSTIC_MSG_SEVERITY** è impostata su 1 per un processo IBM MQ , ciò fa sì che la severità del messaggio venga accodata al numero del messaggio come un singolo carattere alfabetico maiuscolo quando il processo IBM MQ scrive un messaggio in un log degli errori o nella console.

Il comportamento abilitato da **AMQ_DIAGNOSTIC_MSG_SEVERITY** è impostato per impostazione predefinita. Puoi disattivare questo comportamento impostando la variabile di ambiente su 0.

Per ulteriori informazioni, vedi [Utilizzo dei log degli errori](#).

AMQ_DISABLE_CLIENT_AMS

È possibile utilizzare la variabile di ambiente **AMQ_DISABLE_CLIENT_AMS** per disabilitare IBM MQ Advanced Message Security (AMS) sul client se viene riportato un errore 2085 (`MQRC_UNKNOWN_OBJECT_NAME`) quando si tenta di connettersi a un gestore code da una versione precedente del prodotto e si sta utilizzando uno dei seguenti client:

- Un Java runtime environment (JRE) diverso da IBM Java runtime environment (JRE)

- Un client IBM MQ IBM MQ classes for JMS o IBM MQ classes for Java .

Nota: Non è possibile utilizzare la variabile di ambiente **AMQ_DISABLE_CLIENT_AMS** per client C. È necessario utilizzare la variabile di ambiente **MQS_DISABLE_ALL_INTERCEPT** .

Per ulteriori informazioni, consultare [Disabilitazione di Advanced Message Security sul client](#).

AMQ_DMPMQCFG_QSGDISP_DEFAULT

Le richieste sulla disposizione di un gestore code utilizzate dal comando **dmpmqc fg** interrogano solo le definizioni QSGDISP (QMGR) per impostazione predefinita. È possibile richiedere ulteriori definizioni utilizzando la variabile di ambiente **AMQ_DMPMQCFG_QSGDISP_DEFAULT** , che può essere impostata su uno dei seguenti valori:

ATTIVO

Includere solo gli oggetti definiti con QSGDISP (QMGR) o QSGDISP (COPY).

TUTTO

Includere gli oggetti definiti con QSGDISP (QMGR) e QSGDISP (COPY). Se il gestore code è membro di un gruppo di condivisione code, vengono inclusi anche QSGDISP (GROUP) e QSGDISP (SHARED).

Copia

Includi solo, oggetti definiti con QSGDISP (COPY)

GRUPPO

Includere solo oggetti definiti con QSGDISP (GROUP); il gestore code di destinazione deve essere un membro di un gruppo di condivisione code.

QMGR

Includere solo gli oggetti definiti con QSGDISP (QMGR). Questo è il comportamento predefinito se si utilizza questa variabile di ambiente, per corrispondere al comportamento esistente di **dmpmqc fg**.

PRIVATO

Includere solo gli oggetti definiti con QSGDISP (QMGR) o QSGDISP (COPY).

CONDIVISO

Includere solo gli oggetti definiti con QSGDISP (SHARED).

AMQ_IODELAY, AMQ_IODELAY_INMS e AMQ_IODELAY_FFST



IBM MQ rileva quando le operazioni di lettura e scrittura del log o di input e output impiegano più tempo del previsto. Ciò potrebbe essere dovuto a problemi con il sistema operativo o il sistema di memoria e può influire sulle prestazioni del gestore code. Da IBM MQ 9.4.0, è possibile utilizzare le variabili di ambiente **AMQ_IODELAY** per ottimizzare la diagnostica e i tempi quando l'input e l'output per il log del gestore code e il file system di archiviazione sono lenti. Se viene visualizzato il messaggio [AMQ6729W](#) L 'operazione I/O del log ha superato la soglia nel log degli errori del gestore code, esaminare la causa e apportare le modifiche di conseguenza. Utilizzare le variabili come mostrato nei seguenti esempi:

MQ_IODELAY

Tempo di soglia in secondi, il valore predefinito è 1 secondo. Se un'operazione I/O impiega più tempo di questa soglia, nei file di log di IBM MQ viene riportato il messaggio di errore AMQ6729W . Il messaggio di avvertenza viene ripetuto al massimo ogni 10 secondi se i ritardi continuano. È possibile aumentare questo valore per eliminare gli errori o diminuire per esaminare specifici problemi di prestazioni. Ad esempio:

```
export AMQ_IODELAY=200000
```

INM_AMQ_IODELAY_INM

Modificare la misura temporale in microsecondi invece che in secondi. Utilizzare questa opzione per impostare una soglia inferiore prima di ricevere il messaggio AMQ6729 nel log del gestore code.

```
export AMQ_IODELAY_INMS=YES
```

AMQ_IODELAY_FFST

Oltre al messaggio di avvertenza nel log degli errori, ogni volta che viene superata la soglia viene generato un file FFST contenente informazioni diagnostiche.

```
export AMQ_IODELAY_FFST=YES
```

L'avvio del gestore code come in questo esempio, causa la scrittura di un file FDC o FFST se un'operazione di input / output impiega più di 200000 microsecondi (0.2s), che è ancora una soglia relativamente generosa.

Per ulteriori informazioni, fare riferimento alla sezione [Funzionamento del controllo di integrità del gestore code](#).

TRACCIA AMQ_LDAP_

Se la variabile di ambiente **AMQ_LDAP_TRACE** è impostata su un valore non null, è possibile attivare e disattivare la traccia del client LDAP senza arrestare o avviare anche il gestore code.

Per ulteriori informazioni, consultare [Abilitazione della traccia dinamica del codice della libreria client LDAP](#).

METRICA_LICENZA_AMQ

Multi

L'impostazione della variabile di ambiente **AMQ_LICENSING_METRIC=VPCMonthlyPeak** fa in modo che il gestore code carichi i dati relativi a tipi di licenza VPC mensili, invece del comportamento predefinito di caricamento dei dati relativi alle licenze orarie basate sul contenitore.

Per ulteriori informazioni sulla configurazione di IBM MQ per l'utilizzo con il servizio di misurazione IBM Cloud Private , consultare [IBM Cloud Private servizio di misurazione](#) nella documentazione di IBM Cloud Private .

MQ_MQS_INI_LOCATION

Linux

AIX

Su sistemi AIX and Linux , è possibile modificare l'ubicazione utilizzata per il file `mqs.ini` impostando l'ubicazione del file `mqs.ini` nella variabile di ambiente **AMQ_MQS_INI_LOCATION** . Questa variabile di ambiente deve essere impostata a livello di sistema.

Per ulteriori informazioni sul file `mqs.ini` , incluse le ubicazioni delle directory, consultare il file di configurazione [IBM MQ , mqs.ini](#).

FDCS - NO_AMQ_BAD_COMMS_DATA_

La variabile di ambiente **AMQ_NO_BAD_COMMS_DATA_FDCS** è valida quando è impostata su qualsiasi valore.

Se IBM MQ non riconosce la trasmissione iniziale dei dati durante il tentativo di connessione di un client nonIBM MQ a un listener TCP/IP IBM MQ , il gestore code scriverà un messaggio di errore **AMQ9207E** nei log di errori del gestore code. Viene scritto anche un record FDC (failure data capture). È possibile sopprimere la generazione di questi file diagnostici con la variabile di ambiente **AMQ_NO_BAD_COMMS_DATA_FDCS** . Quando **AMQ_NO_BAD_COMMS_DATA_FDCS** è impostato su un valore (ad esempio, TRUE), questo indica a IBM MQ di non generare FFST quando si riportano **AMQ9207E** messaggi di errore sul flusso di comunicazioni iniziale. Per essere efficace, la variabile di ambiente deve essere impostata prima di avviare i processi del gestore code e del listener.

L'FDC continua ad essere generato nel caso in cui un client invii flussi di protocollo IBM MQ validi al gestore code e quindi invii dati non validi, poiché ciò è indicativo di un problema del client che richiede ulteriori indagini.

Nota: L'acquisizione di FFST durante la notifica dei messaggi di errore [AMQ9207E](#) sui flussi di comunicazione iniziali viene eliminata per impostazione predefinita.

AMQ_NO_IPV6

La variabile di ambiente **AMQ_NO_IPV6** è valida quando è impostata su qualsiasi valore. Quando questa variabile di ambiente è impostata, disabilita l'utilizzo di IPv6 durante il tentativo di connessione.

ORDINE_ANNULLAMENTO_COMMIT

La variabile di ambiente **AMQ_REVERSE_COMMIT_ORDER** configura un gestore code in modo che in una transazione XA venga eseguito il commit della modifica del gestore code IBM MQ una volta completato l'aggiornamento del corrispondente database. Le applicazioni che leggono i messaggi dalle code visualizzano un messaggio solo dopo che è stato completato l'aggiornamento del database corrispondente.

Nota: Non impostare **AMQ_REVERSE_COMMIT_ORDER** senza leggere e comprendere lo scenario descritto in [Livello di isolamento](#).

AMQ_SSL_ALLOW_DEFAULT_CERT

Quando la variabile di ambiente **AMQ_SSL_ALLOW_DEFAULT_CERT** non è impostata, un'applicazione può connettersi a un gestore code con un certificato personale nel keystore client solo quando il certificato include il nome etichetta *ibmwebspheremuserid*. Quando la variabile di ambiente **AMQ_SSL_ALLOW_DEFAULT_CERT** è impostata, il certificato non richiede il nome etichetta *ibmwebspheremuserid*. In altre parole, il certificato utilizzato per connettersi a un gestore code può essere un certificato predefinito, purché sia presente un certificato predefinito nel repository delle chiavi e il repository delle chiavi non contenga un certificato personale con prefisso *ibmwebspheremuserid*.

Un valore di 1 abilita l'utilizzo di un certificato predefinito.

Invece di utilizzare la variabile di ambiente **AMQ_SSL_ALLOW_DEFAULT_CERT**, un'applicazione può utilizzare l'impostazione **CertificateLabel** della stanza SSL nel file `mqclient.ini`. Per ulteriori informazioni, consultare [Etichette di certificati digitali, che comprendono i requisiti](#) e la stanza [SSL del file di configurazione del client](#).

AMQ_SSL_LDAP_SERVER_VERSION

La variabile di ambiente **AMQ_SSL_LDAP_SERVER_VERSION** può essere utilizzata per garantire che LDAP v2 o LDAP v3 venga utilizzato dai componenti crittografici IBM MQ nei casi in cui i server CRL richiedono l'utilizzo di una versione specifica del protocollo LDAP.

Impostare la variabile di ambiente sul valore appropriato nell'ambiente utilizzato per avviare il gestore code o il canale:

- Per richiedere l'utilizzo di LDAP v2, impostare `AMQ_SSL_LDAP_SERVER_VERSION=2`.
- Per richiedere l'utilizzo di LDAP v3, impostare `AMQ_SSL_LDAP_SERVER_VERSION=3`.

Questa variabile di ambiente non influisce sulle connessioni LDAP stabilite dal gestore code IBM MQ per l'autenticazione utente o l'autorizzazione utente.

ZLIBNX UTENTE AMQ_



Su AIX, la variabile di ambiente **AMQ_USE_ZLIBNX** può essere utilizzata per abilitare gli agent MCA (message channel agent) per utilizzare la libreria zlibNX accelerata dall'hardware per la compressione e la decompressione dei dati del messaggio quando si utilizzano le tecniche ZLIBFAST o ZLIBHIGH.

Suggerimento: I messaggi altamente comprimibili che hanno una dimensione superiore a 2 KB hanno maggiori probabilità di beneficiare dell'opzione di utilizzare la libreria zlibNX , riducendo l'utilizzo della CPU.

La libreria zlibNX è disponibile in IBM AIX 7.2 con Technology Level 4 Expansion Pack e versioni successive. Se la variabile di ambiente è impostata e la libreria zlibNX (/usr/opt/zlibNX/lib/libz.a) non viene installata, gli agent del canale dei messaggi utilizzeranno la libreria zlib standard fornita nell'installazione di IBM MQ for AIX .

HOME



In AIX, Linux e IBM i, la variabile di ambiente **HOME** specifica il nome della directory in cui viene ricercato il file `mqclient.ini` . Questo file contiene informazioni di configurazione utilizzate da IBM MQ MQI clients.

Per ulteriori informazioni, consultare [IBM MQ MQI client configuration file, mqclient.ini](#) e [Location of the client configuration file](#).

HOMEDRIVE e HOMEPATH



Per essere utilizzate, devono essere impostate sia le variabili di ambiente **HOMEDRIVE** che **HOMEPATH** . Vengono utilizzati su sistemi Windows per specificare il nome della directory in cui viene ricercato il file `mqclient.ini` . Questo file contiene informazioni di configurazione utilizzate da IBM MQ MQI clients.

Per ulteriori informazioni, consultare [IBM MQ MQI client configuration file, mqclient.ini](#) e [Location of the client configuration file](#).

LDAP_BASEDN

LDAP_BASEDN è la variabile di ambiente richiesta per l'esecuzione di un programma di esempio LDAP. Specifica il DN (Distinguished Name) di base per la ricerca nell'indirizzario.

HOST_LDAP

LDAP_HOST è una variabile di ambiente facoltativa per l'esecuzione di un programma di esempio LDAP. Specifica il nome dell'host su cui è in esecuzione il server LDAP; se non viene specificato, viene utilizzato per impostazione predefinita l'host locale.

VERSIONE LDAP

LDAP_VERSION è una variabile di ambiente facoltativa per l'esecuzione di un programma di esempio LDAP. Specifica la versione del protocollo LDAP da utilizzare e può essere 2 o 3. La maggior parte dei server LDAP ora supporta la versione 3 del protocollo; tutti supportano la versione precedente 2. Questo esempio funziona ugualmente bene con entrambe le versioni del protocollo e, se non viene specificato, il valore predefinito è la versione 2.

INTERVALLO_ELIMINAZIONE_CANALE_MQ

La variabile di ambiente **MQ_CHANNEL_SUPPRESS_INTERVAL** specifica l'intervallo di tempo, in secondi, durante il quale i messaggi definiti con **MQ_CHANNEL_SUPPRESS_MSGS** devono essere eliminati dalla scrittura nel log degli errori, insieme al numero di volte in cui un messaggio può verificarsi durante l'intervallo di tempo specificato prima di essere eliminato. Il valore predefinito è 60,5, il che significa che ogni ulteriore ricorrenza di un determinato messaggio viene soppressa dopo le prime cinque ricorrenze di tale messaggio in un intervallo di 60 secondi. Per ulteriori informazioni, consultare [Soppressione dei messaggi di errore del canale dai log degli errori su Multiplatforms](#).

La variabile di ambiente **MQ_CHANNEL_SUPPRESS_INTERVAL** è paragonabile a [SuppressInterval](#) nel file `qm.ini` .

MQ_CHANNEL_SUPPRESS_MSGS

La variabile di ambiente **MQ_CHANNEL_SUPPRESS_MSGS** elimina i messaggi di errore del canale nel log degli errori. È possibile specificare un elenco di messaggi eliminati. **MQ_CHANNEL_SUPPRESS_MSGS** viene utilizzato insieme a **MQ_CHANNEL_SUPPRESS_INTERVAL**, che specifica il numero di volte in cui ogni messaggio viene visualizzato prima di essere eliminato e l'intervallo di tempo per cui i messaggi vengono eliminati. Per ulteriori informazioni, consultare [Soppressione dei messaggi di errore del canale dai log degli errori su Multiplatforms](#).

la variabile di ambiente **MQ_CHANNEL_SUPPRESS_MSGS** è paragonabile a `SuppressMessage` nel file `qm.ini`, tranne che è possibile eliminare qualsiasi messaggio del canale utilizzando la variabile di ambiente, mentre esiste un elenco restrittivo per il metodo `qm.ini`.

TIPO_MQ_CONNECT_



Su Multiplatforms, è possibile utilizzare la variabile d'ambiente **MQ_CONNECT_TYPE** in combinazione con il tipo di collegamento specificato nel campo Opzioni della struttura MQCNO utilizzata su una chiamata MQCONN. **MQ_CONNECT_TYPE** ha effetto solo per i bind STANDARD. Per altri bind, **MQ_CONNECT_TYPE** viene ignorato.

Per ulteriori informazioni, consultare [Utilizzo delle opzioni di chiamata MQCONN con MQ_CONNECT_TYPE](#).

MQ_CROSS_QUEUE_ORDER_ALL

Quando si imposta la variabile di ambiente **MQ_CROSS_QUEUE_ORDER_ALL** su un valore diverso da zero, l'ordine di inserimento del messaggio viene conservato in un'unità di lavoro. Ciò significa che, se i messaggi in una UOW (Unit of Work) (UoW) vengono inseriti in più code (ad esempio, Q1, quindi Q2), quando viene emesso un MQCMIT, i messaggi vengono consegnati e resi disponibili nello stesso ordine della coda in cui erano PUT.

In un ambiente con più gestori code, **MQ_CROSS_QUEUE_ORDER_ALL** deve esistere e avere un valore non vuoto sia sul lato di invio che su quello di ricezione prima che ogni gestore code venga avviato.

MQ_EPHEMER_PREFIX

La variabile di ambiente **MQ_EPHEMERAL_PREFIX** specifica il percorso della directory temporanea del gestore code, all'interno della quale vengono conservati i dati del gestore code temporaneo, mentre il gestore code è in esecuzione.

Come alternativa alla modifica del prefisso temporaneo modificando l'attributo **EphemeralPrefix** nell'attributo **DefaultEphemeralPrefix** della stanza AllQueueManagers del file `mqs.ini`, è possibile utilizzare la variabile d'ambiente **MQ_EPHEMERAL_PREFIX** per sovrascrivere **EphemeralPrefix** per il comando `crtmqm`. Per ulteriori informazioni, consultare [Configurable ephemeral directory](#).

PERCORSO MQ_FILE



La variabile di ambiente **MQ_FILE_PATH** viene configurata durante l'installazione dei pacchetti di runtime sulla piattaforma Windows. Questa variabile di ambiente contiene gli stessi dati della seguente chiave nel registro Windows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

Per ulteriori informazioni, consultare [setmqenv \(set IBM MQ environment\)](#) e [crtmqenv \(create IBM MQ environment\)](#).

MQ_JAVA_DATA_PATH

La variabile di ambiente **MQ_JAVA_DATA_PATH** specifica la directory per l'output di log e traccia per IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java. Viene utilizzato dagli script forniti con IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java.

Per ulteriori informazioni, consultare [Impostazione delle variabili di ambiente per le classi IBM MQ per JMS/Jakarta Messaging](#) e [Variabili di ambiente relative alle classi IBM MQ per Java](#).

PERCORSO_INSTALL_JAVA_MQ_

La variabile di ambiente **MQ_JAVA_INSTALL_PATH** specifica la directory in cui sono installati IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging come mostrato in [Elementi installati per le classi IBM MQ per JMS](#) e IBM MQ classes for Java come mostrato in [Directory di installazione IBM MQ classes for Java](#).

Per ulteriori informazioni, consultare [Impostazione delle variabili di ambiente per le classi IBM MQ per JMS/Jakarta Messaging](#) e [Variabili di ambiente relative alle classi IBM MQ per Java](#).

MQ_JAVA_LIB_PATH

La variabile di ambiente **MQ_JAVA_LIB_PATH** specifica la directory in cui sono memorizzate le librerie IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging e IBM MQ classes for Java. Alcuni script, ad esempio IVTRun, forniti con IBM MQ classes for JMS e IBM MQ classes for Jakarta Messaging o IBM MQ classes for Java utilizzano questa variabile di ambiente.

Per ulteriori informazioni, consultare [Impostazione delle variabili di ambiente per le classi IBM MQ per JMS/Jakarta Messaging](#) e [Variabili di ambiente relative alle classi IBM MQ per Java](#).

MQ_OVERRIDE_DATA_PATH

È possibile utilizzare la variabile di ambiente **MQ_OVERRIDE_DATA_PATH** per modificare la directory predefinita del percorso dati IBM MQ.

MQ_SET_NODELAYACK

AIX

La variabile di ambiente **MQ_SET_NODELAYACK** disattiva il riconoscimento ritardato TCP su AIX.

Quando si imposta questa variabile di ambiente, l'impostazione disattiva la ricezione ritardata TCP richiamando la chiamata `setsockopt` del sistema operativo con l'opzione `TCP_NODELAYACK`. Solo AIX supporta questa funzione, quindi la variabile di ambiente **MQ_SET_NODELAYACK** ha effetto solo su AIX.

MQ_XX_ENCODE_CASE_ONE nome_utente

Linux

È possibile utilizzare la variabile di ambiente **MQ_USER_NAME** per consentire a un'installazione non registrata su Linux di scegliere il nome di un utente senza nome. Ciò è necessario, ad esempio, per utilizzare le gerarchie di pubblicazione / sottoscrizione in OpenShift.

Il valore di **MQ_USER_NAME** non deve corrispondere a un utente preesistente sul sistema e deve essere inferiore o uguale a 12 byte.

LOGFILE MQAPI_TRACE_

Il programma di uscita API di esempio genera una traccia MQI per un file specificato dall'utente con un prefisso che è definito nella variabile di ambiente **MQAPI_TRACE_LOGFILE**.

Per ulteriori informazioni, consultare [Il programma di esempio dell'uscita API](#).

NOME MQAPPL

ALW

Se il nome dell'applicazione non è stato ancora scelto, è possibile utilizzare la variabile di ambiente **MQAPPLNAME** come nome da utilizzare per identificare la connessione a un gestore code. Vengono utilizzati solo i primi 28 caratteri e non devono essere tutti spazi o valori null.

Per ulteriori informazioni, consultare [Utilizzo del nome applicazione nei linguaggi di programmazione supportati](#).

MQCCSID

La variabile di ambiente **MQCCSID** specifica il numero della serie di caratteri codificati da utilizzare e sostituisce il valore CCSID con cui è stato configurato il server. **MQCCSID** può essere utilizzato per sovrascrivere il CCSID nativo di un'applicazione e specificare il numero della serie di caratteri codificati da utilizzare, ad esempio se il CCSID nativo è un CCSID non supportato o non è il CCSID richiesto.

Per impostare **MQCCSID**, utilizzare uno dei seguenti comandi:

- Linux AIX Su AIX and Linux:

```
export MQCCSID=number
```

- Windows Su Windows:

```
SET MQCCSID=number
```

- IBM i Su IBM i:

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

Per ulteriori informazioni, consultare [Scelta del CCSID del client o del server](#).

URL MQCCDT

La variabile di ambiente **MQCCDTURL** fornisce la capacità equivalente di impostare una combinazione delle variabili di ambiente **MQCHLLIB** e **MQCHLTAB**. Consente di fornire un file, ftp o URL http come un singolo valore da cui è possibile ottenere una tabella di definizione del canale client per i programmi nativi che si collegano come client, ovvero applicazioni C, COBOL o C++.

Nota: L'utilizzo delle variabili di ambiente per fornire l'URL non ha alcun effetto per le applicazioni Java, JMS o .NET gestite.

IBM MQ supporta il richiamo di una CCDT da un file, ftp o URL http. Tuttavia, **MQCCDTURL** accetta solo un valore URL. Non accetta il formato indirizzario del file system locale esistente.

Per utilizzare **MQCCDTURL** al posto di **MQCHLLIB** e **MQCHLTAB** con un file locale, puoi utilizzare un protocollo 'file://'. Pertanto, come mostrato in questo esempio per AIX e Linux:

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

equivale a:

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

Puoi specificare anche un file JSON come mostrato in questo esempio per Windows:

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```


equivale a:

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

Per ulteriori informazioni, consultare [Accesso URL a CCDT](#).

MQCERTLABL

La variabile di ambiente di **MQCERTLABL** definisce l'etichetta del certificato di una definizione di canale che IBM MQ deve utilizzare per individuare un certificato personale inviato durante un handshake TLS.

Per ulteriori informazioni, consultare [Etichette di certificati digitali, che comprendono i requisiti](#).

MQCERTVPOL

La variabile di ambiente **MQCERTVPOL** specifica il tipo di politica di convalida del certificato da utilizzare. Questa variabile di ambiente sovrascrive l'attributo **CertificateValPolicy** nella stanza SSL del file di configurazione client.

MQCERTVPOL può essere impostato su uno dei seguenti due valori:

ANY

Utilizzare qualsiasi politica di convalida del certificato supportata dalla libreria dei socket protetti sottostante. Questa è l'impostazione predefinita.

RFC5280

Utilizzare solo la convalida del certificato conforme allo standard RFC 5280.

Per impostare **MQCERTVPOL**, utilizzare uno dei seguenti comandi:

- Linux AIX Per sistemi AIX and Linux :

```
export MQCERTVPOL= value
```

- Windows Per sistemi Windows :

```
SET MQCERTVPOL= value
```

- IBM i Per sistemi IBM i :

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

Per ulteriori informazioni, consulta [Certificate validation policies in IBM MQ](#) e [Configuring certificate validation policies in IBM MQ](#).

MQCHLLIB

La variabile di ambiente **MQCHLLIB** specifica il percorso di directory del file che contiene la CCDT (client channel definition table). Il file viene creato sul server, ma può essere copiato sulla workstation IBM MQ MQI client .

Per impostare **MQCHLLIB**, utilizzare uno dei seguenti comandi:

- Windows Su Windows:

```
SET MQCHLLIB=pathname
```

Ad esempio:

```
SET MQCHLLIB=C:\wmqtest
```

- **Linux** **AIX** Per sistemi AIX and Linux :

```
export MQCHLLIB=pathname
```

- **IBM i** Per IBM i:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

Se **MQCHLLIB** non è impostato, il percorso per il client assume il valore predefinito:

- **Linux** **AIX** Su AIX and Linux: `/var/mqm/`
- **Windows** Su Windows: `MQ_INSTALLATION_PATH`
- **IBM i** Su IBM i: `/QIBM/UserData/mqm/`

Per i comandi **crtmqm** e **strmqm**, il percorso assume il valore predefinito di una delle due serie di percorsi. Se *datapath* è impostato, il valore predefinito del percorso è uno della prima serie. Se *datapath* non è impostato, il percorso assume il valore predefinito di uno della seconda serie.

- **Linux** **AIX** Su AIX and Linux: `datapath/@ipcc`
- **Windows** Su Windows: `datapath\@ipcc`
- **IBM i** Su IBM i: `datapath/&ipcc`

Oppure:

- **Linux** **AIX** Su AIX and Linux: `/prefix/qmgrs/qmgrname/@ipcc`
- **Windows** Su Windows: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`
- **IBM i** Su IBM i: `/prefix/qmgrs/qmgrname/&ipcc`

dove:

- `MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ .
- Se presente, *datapath* è il valore di DataPath definito nella stanza del gestore code.
- *prefix* è il valore del prefisso definito nella stanza del gestore code. Il prefisso è di solito uno dei seguenti valori:
 - **Linux** **AIX** `/var/mqm` su sistemi AIX and Linux .
 - **IBM i** `/QIBM/UserData/mqm/` su IBM i.
- *qmgrname* è il valore dell'attributo Directory definito nella stanza del gestore code. Il valore potrebbe essere diverso dal nome del gestore code effettivo. Il valore potrebbe essere stato modificato per sostituire i caratteri speciali.
- La posizione in cui è definita la sezione del gestore code dipende dalla piattaforma:
 - **IBM i** **Linux** **AIX** Nel file `mqc.ini` su IBM i, AIX and Linux.
 - **Windows** Nel registro su Windows.

Note:

1. **z/OS** Se si utilizza IBM MQ for z/OS come server, il file deve essere conservato sulla workstation client IBM MQ .

2. Se impostato, MQCHLLIB sovrascrive il percorso utilizzato per individuare la CCDT.
3. MQCHLLIB può contenere un URL che funziona in combinazione con la variabile di ambiente MQCHLTAB (consultare “Accesso URL alla CCDT” a pagina 54).
4. Le variabili di ambiente, come ad esempio **MQCHLLIB**, possono essere indirizzate a un processo o a un lavoro o a livello di sistema, in un modo specifico per la piattaforma.
5. Se si imposta **MQCHLLIB** a livello di sistema su un server, questo imposta lo stesso percorso al file CCDT per tutti i gestori code sul server. Se non si imposta la variabile di ambiente **MQCHLLIB**, il percorso è diverso per ciascun gestore code. I gestori code leggono il valore di **MQCHLLIB**, se impostato, sul comando **crtmqm** o **strmqm**.
6. Se si creano più gestori code su un server, la distinzione è importante, per il seguente motivo. Se si imposta **MQCHLLIB** a livello di sistema, ogni gestore code aggiorna lo stesso file CCDT. Il file contiene le definizioni di connessione client da tutti i gestori code sul server. Se la stessa definizione esiste su più gestori code, ad esempio SYSTEM.DEF.CLNTCONN, il file contiene la definizione più recente. Quando si crea un gestore code, se **MQCHLLIB** è impostato, SYSTEM.DEF.CLNTCONN viene aggiornato in CCDT. L'aggiornamento sovrascrive il SYSTEM.DEF.CLNTCONN creato da un gestore code differente. Se è stata modificata la definizione precedente, le modifiche vengono perse. Per questo motivo, è necessario considerare la ricerca di alternative all'impostazione di **MQCHLLIB** come variabile di ambiente a livello di sistema sul server.
7. L'opzione NOREPLACE MQSC e PCF su una definizione di connessione client non controlla il contenuto del file CCDT. Una definizione di canale di connessione client con lo stesso nome precedentemente creato, ma non da questo gestore code, viene sostituita, indipendentemente dall'opzione NOREPLACE. Se la definizione è stata precedentemente creata dallo stesso gestore code, la definizione non viene sostituita.
8. Il comando **rcrmqobj -t c1chl1tab** elimina e ricrea il file CCDT. Il file viene ricreato solo con le definizioni di connessioni client create sul gestore code su cui è in esecuzione il comando.
9. Altri comandi che aggiornano CCDT modificano solo i canali di connessione client che hanno lo stesso nome di canale. Altri canali di connessione client nel file non vengono modificati.
10. Il percorso per **MQCHLLIB** non necessita di virgolette.

Per ulteriori informazioni, consultare [Ubicazioni per CCDT](#), [Accesso URL a CCDT](#) e [Connessione delle applicazioni client ai gestori code utilizzando le variabili di ambiente](#).

MQCHLTAB

La variabile di ambiente **MQCHLTAB** specifica il nome del file che contiene la CCDT (client channel definition table). Il nome file predefinito è AMQCLCHL.TAB.


Per impostare **MQCHLTAB**, utilizzare uno dei seguenti comandi:

-   Su AIX and Linux:

```
export MQCHLTAB=filename
```

-  Su Windows:

```
SET MQCHLTAB=filename
```

-  Su IBM i:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

Ad esempio:

```
SET MQCHLTAB=ccdf1.tab
```

Come per il client, la variabile di ambiente **MQCHLTAB** sul server specifica il nome della tabella di definizione del canale client.

Per ulteriori informazioni, consultare [Ubicazioni per CCDT](#), [Accesso URL a CCDTe](#) [Connessione delle applicazioni client ai gestori code utilizzando le variabili di ambiente](#).

MQCLNTCF

La variabile di ambiente **MQCLNTCF** specifica l'ubicazione del file di configurazione IBM MQ MQI client . Questo file contiene informazioni di configurazione utilizzate da IBM MQ MQI clients.

È possibile utilizzare la variabile di ambiente **MQCLNTCF** per modificare il percorso del file `mqclient.ini`.

Il formato di questa variabile di ambiente è un URL completo. Ciò significa che il nome file potrebbe non essere necessariamente `mqclient.ini`, il che facilita l'inserimento del file su un file system collegato in rete. Per ulteriori informazioni, consultare [IBM MQ MQI client configuration file, mqclient.ini e Location of the client configuration file](#).

TRACE_MQDOTNET_ATTIVO

La variabile di ambiente **MQDOTNET_TRACE_ON** viene utilizzata per abilitare la traccia per client IBM MQ .NET ridistribuibili. I valori uguali e inferiori a 0 non abilitano la traccia, 1 abilita la traccia predefinita e i valori superiori a 1 abilitano la traccia dei dettagli.

Per ulteriori informazioni, consultare [Traccia delle applicazioni IBM MQ .NET](#) e [Traccia delle applicazioni IBM MQ .NET utilizzando le variabili di ambiente](#).

MQIPADDRV

La variabile di ambiente **MQIPADDRV** specifica quale protocollo IP utilizzare per una connessione del canale. Ha i valori stringa possibili di "MQIPADDR_IPV4" o "MQIPADDR_IPV6". Questi valori hanno lo stesso significato di IPv4 e IPv6 in [ALTER QMGR IPADDRV](#) e l'attributo **IPAddressVersion** della stanza TCP del file di configurazione client. Se la variabile di ambiente non è impostata, viene utilizzato "MQIPADDR_IPV4".

Per impostare **MQIPADDRV**, utilizzare uno dei seguenti comandi:

-   Su AIX and Linux:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6"/>
```

-  Su Windows:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

-  Su IBM i:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPV4|MQIPADDR_IPV6)
```

MQKEYRPWD

Quando imposti la variabile di ambiente **MQKEYRPWD**, specifica la password per il repository delle chiavi che contiene il certificato digitale appartenente all'utente. Se si utilizza **MQKEYRPWD**, è necessario codificare la password prima di impostare il valore della variabile di ambiente.

Per impostare **MQKEYRPWD**, utilizzare uno dei seguenti comandi:

- **Linux** **AIX** Su sistemi AIX and Linux:

```
export MQKEYRPWD=passphrase
```

- **Windows** Su sistemi Windows:

```
SET MQKEYRPWD=passphrase
```

- **IBM i** Su IBM i:

```
ADDENVVAR ENVVAR(MQKEYRPWD) VALUE(passphrase)
```

Non esiste alcun valore predefinito per questa variabile di ambiente.

Per ulteriori informazioni, consultare

- **ALW** [Fornire la password del repository delle chiavi per un IBM MQ MQI client su AIX, Linux, and Windows e Codifica della password del repository delle chiavi](#)
- **IBM i** [Fornire la password del repository delle chiavi per un IBM MQ MQI client su IBM i e Codifica della password del repository delle chiavi.](#)

MQLICENSE

Linux

Sui sistemi Linux , è possibile utilizzare la variabile di ambiente **MQLICENSE** per accettare o visualizzare una licenza IBM MQ dopo aver installato il prodotto.

Per ulteriori informazioni sul motivo per cui si desidera o è necessario eseguire questa operazione, consultare [Accettazione della licenza su IBM MQ per Linux](#)

La variabile di ambiente **MQLICENSE** può essere impostata su uno dei due seguenti valori:

accetta

Accettare la licenza post - installazione.

vista

Visualizzare la licenza, se la licenza è stata accettata.

Per accettare la post - installazione della licenza, utilizzare questo comando:

```
export MQLICENSE=accept
```

Per visualizzare la licenza, utilizzare questo comando:

```
export MQLICENSE=view
```

Nota: È inoltre possibile utilizzare i seguenti comandi per accettare e visualizzare la licenza:

- [mqlicense](#) (accetta la licenza dopo l'installazione)
- [dspmqlic](#) (visualizza licenza IBM MQ)

DIMENSIONE MQMAXERRORLOGSIZE

Multi

La variabile di ambiente **MQMAXERRORLOGSIZE** specifica la dimensione del file di log degli errori del gestore code copiato nel backup.

Per ulteriori informazioni, vedi [Utilizzo dei log degli errori.](#)

NOME

Windows

La variabile di ambiente **MQNAME** specifica il nome NetBIOS locale che i processi IBM MQ possono utilizzare. Una connessione NetBIOS si applica solo a un client e a un server su cui è in esecuzione Windows.

Per impostare **MQNAME**, utilizzare questo comando:

```
SET MQNAME=Your_env_Name
```

Ad esempio:

```
SET MQNAME=CLIENT1
```

Alcune implementazioni NetBIOS richiedono un nome univoco, impostato da **MQNAME**, per ogni applicazione se si eseguono più applicazioni IBM MQ contemporaneamente su IBM MQ MQI client.

Per ulteriori informazioni, consultare [Defining the IBM MQ local NetBIOS name](#).

MQNOREMPOOL

Quando si imposta la variabile di ambiente **MQNOREMPOOL**, si disattiva il pool di canali e i canali vengono eseguiti come thread del listener.

Per ulteriori informazioni, fare riferimento a [MCATYPE \(Message channel agent type\)](#).

LOGFILE MQPSE_TRACE_

Si utilizza la variabile di ambiente **MQPSE_TRACE_LOGFILE** quando si esegue il programma di esempio di uscita di pubblicazione AMQSPSE0, che è un esempio di programma C di un'uscita per intercettare una pubblicazione prima che venga consegnata a un sottoscrittore. Nel processo dell'applicazione da tracciare, questa variabile di ambiente descrive dove devono essere scritti i file di traccia.

Per ulteriori informazioni, consultare [Programma di esempio Pubblica uscita](#).

MQS_AMSCRED_XX_ENCODE_CASE_ONE file_chiave

È possibile utilizzare la variabile di ambiente **MQS_AMSCRED_KEYFILE** per sovrascrivere o fornire il file di chiavi iniziale da utilizzare al runtime delle applicazioni IBM MQ Advanced Message Security (AMS) o quando si sta proteggendo un file di configurazione keystore utilizzando il comando **runamscred**.

Per ulteriori informazioni, consultare [Utilizzo di keystore e certificati con AMS](#) e [Protezione delle password nei file di configurazione del componente IBM MQ](#).

MQS_DISABLE_ALL_INTERCEPT

È possibile utilizzare la variabile di ambiente **MQS_DISABLE_ALL_INTERCEPT** per disabilitare IBM MQ Advanced Message Security (AMS) se viene riportato un errore 2085 (MQRC_UNKNOWN_OBJECT_NAME) quando si tenta di connettersi a un gestore code da una versione precedente del prodotto e si utilizza IBM MQ con client C nativi.

Nota: È possibile utilizzare la variabile di ambiente **MQS_DISABLE_ALL_INTERCEPT** solo per i client C. Per client Java, è necessario utilizzare invece la variabile di ambiente **AMQ_DISABLE_CLIENT_AMS**.

Per ulteriori informazioni, consultare [Disabilitazione di Advanced Message Security sul client](#).

HOST IP_MQS

Poiché gli oggetti del file system IPC devono essere distinti dal sistema, una sottodirectory per ciascun sistema su cui viene eseguito il gestore code viene aggiunta al percorso della directory. Se il valore generato del nome host crea un problema, è possibile impostare il nome host utilizzando la variabile di ambiente **MQS_IPC_HOST**.

Per ulteriori informazioni, consultare [Condivisione dei file IBM MQ su Multiplatforms](#).

MQS_KEYSTORE_CONF

La variabile di ambiente **MQS_KEYSTORE_CONF** specifica l'ubicazione del file di configurazione del keystore per IBM MQ Advanced Message Security (AMS), se il file non si trova nell'ubicazione predefinita di *home_directory/.mq/keystore.conf*.

Per ulteriori informazioni, consultare [Utilizzo di keystore e certificati con AMS](#).

Se hai dei problemi su Managed File Transfer, vedi [Risoluzione dei problemi quando MFT non legge le proprietà del keystore per AMS](#).

FILE_CHIAVE_MQS

Quando si configura la variabile di ambiente **MQS_MQI_KEYFILE**, specifica l'ubicazione di un file di chiavi iniziale che contiene la chiave iniziale da utilizzare per le operazioni di protezione con password. Se il file di chiavi iniziale non è specificato, la chiave iniziale predefinita viene utilizzata dal IBM MQ sistema di protezione con password.

Per impostare **MQS_MQI_KEYFILE**, utilizzare uno dei seguenti comandi:

- Linux AIX Su sistemi AIX and Linux:

```
export MQS_MQI_KEYFILE=key file location
```

- Windows Su sistemi Windows:

```
SET MQS_MQI_KEYFILE=key file location
```

- IBM i Su IBM i:

```
ADDENVVAR ENVVAR(MQS_MQI_KEYFILE) VALUE(key file location)
```

Per ulteriori informazioni, vedi [Come fornire una chiave iniziale per un IBM MQ MQI client su AIX, Linux, and Windows](#) e [Come fornire una chiave iniziale per un IBM MQ MQI client su IBM i](#).

FILE CHIAVE SSLCRYP MQS

La variabile di ambiente **MQS_SSLCRYP_KEYFILE** rappresenta un modo alternativo per specificare il nome e il percorso completo del file contenente la chiave iniziale utilizzata per codificare la password nella stringa di configurazione hardware crittografica PKCS #11, invece di specificarla con l'attributo **SSLCryptoHardwareKeyFile** nella stanza *SSL* di *qm.ini*. **MQS_SSLCRYP_KEYFILE** ha una priorità superiore rispetto al file *qm.ini*, quindi il suo valore ha la precedenza su qualsiasi altro valore. Per ulteriori informazioni, consultare [ClientIBM MQ che utilizzano hardware crittografico](#).

MQS_TRACE_OPZIONI

AIX

Per la traccia selettiva dei componenti su AIX, utilizzare la variabile d'ambiente **MQS_TRACE_OPTIONS** per attivare singolarmente le funzioni di traccia dei parametri e dei dettagli.

Nota: Impostare la variabile di ambiente **MQS_TRACE_OPTIONS** solo se è stato richiesto dal supporto IBM.

Per ulteriori informazioni, vedi [Traccia su AIX and Linux](#).

SERVER MQT

La variabile di ambiente **MQSERVER** viene utilizzata per definire un canale minimo. **MQSERVER** specifica l'ubicazione del server IBM MQ e il metodo di comunicazione da utilizzare.

Nota: Non puoi utilizzare **MQSERVER** per definire un canale TLS o un canale con uscite canale. Per ulteriori informazioni su come definire un canale TLS, vedi [Protezione dei canali con TLS](#).


I seguenti esempi mostrano come impostare **MQSERVER**:

-   Su AIX and Linux:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

-  Su Windows:

```
SET MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

-  Su IBM i:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

Nota:

- Il nome del canale non può contenere il carattere barra (/) poiché questo carattere viene utilizzato per separare il nome del canale, il tipo di trasporto e il nome della connessione. Quando si utilizza la variabile di ambiente **MQSERVER** per definire un canale client, viene utilizzata una lunghezza massima del messaggio (MAXMSGL) di 100 MB. Pertanto, la dimensione massima del messaggio in vigore per il canale è il valore specificato nel canale SVRCONN sul server.
- Il tipo di trasporto può essere uno tra LU62 , TCP , NETBIOS, SPX, a seconda della piattaforma client IBM MQ .
- Il nome della connessione deve essere un nome di rete completo. ad esempio AMACHINE.ACOMPANY.COM(1414).
- Il nome connessione può essere un elenco separato da virgole di nomi connessione. I nomi di connessione nell'elenco vengono utilizzati in modo simile a più connessioni in una tabella di connessioni client. L'elenco dei nomi delle connessioni potrebbe essere utilizzato come alternativa ai gruppi di gestori code per specificare più collegamenti per il client da provare. Se si sta configurando un gestore code a più istanze, è possibile utilizzare un elenco di nomi di connessione per specificare diverse istanze del gestore code.

Se si utilizza la variabile di ambiente **MQSERVER** per definire il canale tra la macchina IBM MQ MQI client e una macchina server, questo è l'unico canale disponibile per l'applicazione e non viene fatto alcun riferimento alla CCDT (client channel definition table).

Per ulteriori informazioni, consultare [Creazione di un canale di connessione client sul client MQI IBM MQ utilizzando MQSERVER](#).

MQSNOAUT



Avvertenza: Questa funzionalità non è consigliata.

Quando si imposta la variabile di ambiente **MQSNOAUT** su qualsiasi valore, disabilita OAM (object authority manager) e impedisce qualsiasi controllo di sicurezza. Potrebbe essere adatto per un ambiente di test. Ciò include sia l'autorizzazione che la funzionalità di autenticazione della connessione. TLS, i record di autenticazione di canale e le uscite di sicurezza non sono interessati.

La variabile di ambiente **MQSNOAUT** ha effetto solo quando viene creato un gestore code.



Avvertenza: Per abilitare OAM, è necessario eliminare il gestore code, eliminare la variabile di ambiente e ricreare il gestore code senza specificare **MQSNOAUT**.

Per ulteriori informazioni, consultare [Prevenzione dei controlli di accesso di sicurezza sui sistemi AIX, Linux e Windows](#).

MQSPREFIX

In alternativa alla modifica del prefisso predefinito, è possibile utilizzare la variabile di ambiente **MQSPREFIX** per sovrascrivere il **DefaultPrefix** per il comando **crtmqm**.

Per ulteriori informazioni, fare riferimento a [IBM MQ nomi file](#) e alla stanza [AllQueueManagers](#) del file `mqs.ini`.

CCRYMQSSL




La variabile di ambiente **MQSSLCRYP** contiene una stringa di parametro che è possibile utilizzare per configurare l'hardware crittografico presente nel sistema.

I valori consentiti sono uguali a quelli del campo [SSLCryptoHardware](#) nella stanza SSL del file di configurazione client.

Per impostare **MQSSLCRYP**, utilizzare uno dei seguenti comandi:

-   Su sistemi AIX and Linux:

```
export MQSSLCRYP=string
```

-  Su sistemi Windows:

```
SET MQSSLCRYP=string
```

Per ulteriori informazioni, consultare [Configurazione per l'hardware crittografico su AIX, Linux, and Windows e IBM MQ clients che utilizzano l'hardware crittografico in Protezione delle password nei file di configurazione del componente IBM MQ](#).

FIPS MQSSL

La variabile di ambiente **MQSSLFIPS** specifica se devono essere utilizzati solo algoritmi certificati FIPS se la crittografia viene eseguita in IBM MQ. È possibile impostare questa variabile di ambiente su YES o NO. Il valore predefinito è NO. Questi valori sono gli stessi del parametro **SSLFIPS** del comando [ALTER QMGR](#).

Per impostare **MQSSLFIPS**, utilizzare uno dei seguenti comandi:

-   Su sistemi AIX and Linux:

```
export MQSSLFIPS=YES|NO
```

-  Su sistemi Windows:

```
SET MQSSLFIPS=YES|NO
```

- ▶ **IBM i** Su IBM i:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

L'uso di algoritmi certificati FIPS è influenzato dall'uso di hardware crittografico. Per ulteriori informazioni, consultare [Specifica che solo le CipherSpecs certificate FIPS vengono utilizzate al runtime sul client MQI](#).

MQSSLKEYR

La variabile di ambiente **MQSSLKEYR** specifica l'ubicazione del repository chiavi che contiene il certificato digitale appartenente all'utente.

Specificare il percorso completo e il nome file del repository chiavi. Se il suffisso del file non viene specificato, si presuppone che sia `.kdb`.

Per impostare **MQSSLKEYR**, utilizzare uno dei seguenti comandi:

- ▶ **Linux** ▶ **AIX** Su sistemi AIX and Linux:

```
export MQSSLKEYR=pathname
```

- ▶ **Windows** Su sistemi Windows:

```
SET MQSSLKEYR=pathname
```

- ▶ **IBM i** Su IBM i:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

Non esiste alcun valore predefinito per questa variabile di ambiente.

Per ulteriori informazioni, consultare il parametro **SSLKEYR** del comando [ALTER QMGR](#).

MQSSLPROX

La variabile di ambiente **MQSSLPROXY** specifica il nome host e il numero di porta del server proxy HTTP che deve essere utilizzato da GSKit per i controlli OCSP.

Per impostare **MQSSLPROXY**, utilizzare uno dei seguenti comandi:

- ▶ **Linux** ▶ **AIX** Su sistemi AIX and Linux:

```
export MQSSLPROXY="string"
```

- ▶ **Windows** Su sistemi Windows:

```
SET MQSSLPROXY= string
```

La stringa specificata con **MQSSLPROXY** può essere il nome host o l'indirizzo di rete del server proxy HTTP che deve essere utilizzato da GSKit per i controlli OCSP. Questo indirizzo può essere seguito da un numero di porta facoltativo, racchiuso tra parentesi. Se non si specifica alcun numero, viene utilizzata la porta HTTP predefinita (80).

- ▶ **Linux** ▶ **AIX** Ad esempio, su sistemi AIX and Linux, è possibile utilizzare uno dei seguenti comandi:

- ```
export MQSSLPROXY="proxy.example.com(80) "
```
- ```
export MQSSLPROXY="127.0.0.1"
```

Per ulteriori informazioni, consultare [Utilizzo di OCSP \(Online Certificate Status Protocol\)](#).

RESET MQSSL

La variabile di ambiente **MQSSLRESET** specifica il numero di byte non codificati inviati e ricevuti su un canale TLS prima che la chiave segreta TLS venga rinegoziata. Può essere impostato su un numero intero compreso tra 0 e 999 999 999. Il valore predefinito è 0, che indica che le chiavi segrete non vengono mai rinegoziate. Se si specifica un conteggio di reimpostazione della chiave segreta TLS compreso nell'intervallo tra 1 byte e 32 KB, i canali TLS utilizzano un conteggio di reimpostazione della chiave segreta di 32 KB. Questo conteggio di reimpostazioni segrete evita un numero eccessivo di reimpostazioni di chiavi che si verificherebbe per valori di reimpostazione di chiavi segrete TLS di piccole dimensioni.

Per impostare **MQSSLRESET**, utilizzare uno dei seguenti comandi:

-   Su sistemi AIX and Linux:

```
export MQSSLRESET=integer
```

-  Su sistemi Windows:

```
SET MQSSLRESET=integer
```

-  Su IBM i:

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

Per ulteriori informazioni, vedi [Reimpostazione delle chiavi segrete SSL e TLS](#).

MQSUITEB



Puoi configurare IBM MQ per operare in conformità con lo standard NSA Suite B su piattaforme AIX, Linux, and Windows .

La variabile di ambiente **MQSUITEB** specifica se deve essere utilizzata la crittografia conforme a Suite B. Se deve essere utilizzata la crittografia Suite B, è possibile specificare la potenza della crittografia impostando **MQSUITEB** su uno dei seguenti valori:

- Nessuna
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

È possibile specificare più valori utilizzando un elenco separato da virgole. L'utilizzo del valore NONE con qualsiasi altro valore non è valido.

Per ulteriori informazioni, vedi [Configurazione di IBM MQ per Suite B](#).

MQTCPTIMEOUT

La variabile di ambiente **MQTCPTIMEOUT** specifica per quanto tempo IBM MQ attende una chiamata di connessione TCP.

ODQ_MSG

Se si utilizza un gestore code di messaggi non recapitabili diverso da quello di `runmqdlq`, l'origine dell'esempio, `amqsdlq`, è disponibile per l'utilizzo come base. L'esempio è simile al gestore di lettere non recapitate fornito all'interno del prodotto, ma la traccia e la notifica degli errori sono differenti. Utilizzare la variabile di ambiente `ODQ_MSG` per impostare il nome del file contenente i messaggi di errore e di informazioni. Il file fornito è denominato `amqsdlq.msg`.

Per ulteriori informazioni, vedere [Esempio di gestore code di messaggi non instradabili](#).

TRACCIA_ODQ

Se si utilizza un gestore code di messaggi non recapitabili diverso da quello di `runmqdlq`, l'origine dell'esempio, `amqsdlq`, è disponibile per l'utilizzo come base. L'esempio è simile al gestore di lettere non recapitate fornito all'interno del prodotto, ma la traccia e la notifica degli errori sono differenti. Per abilitare la traccia, impostare la variabile di ambiente `ODQ_TRACE` su YES o yes.

Per ulteriori informazioni, vedere [Esempio di gestore code di messaggi non instradabili](#).

WCF_TRACE_ON

Sono disponibili due diversi metodi di traccia per il canale personalizzato WCF. Questi due metodi di traccia vengono attivati indipendentemente o insieme. Ogni metodo produce il proprio file di traccia, quindi quando entrambi i metodi di traccia sono stati attivati, vengono generati due file di output di traccia. Esistono quattro combinazioni per abilitare e disabilitare i due diversi metodi di traccia. Oltre a queste combinazioni per abilitare la traccia WCF, la traccia XMS .NET può essere abilitata utilizzando la variabile di ambiente `WCF_TRACE_ON`.

Per ulteriori informazioni, vedi [Traccia del canale personalizzato WCF per IBM MQ](#).

HOME WMQSOAP

La variabile di ambiente `WMQSOAP_HOME` viene utilizzata quando si completano ulteriori operazioni di configurazione dopo che l'ambiente di hosting del servizio .NET SOAP over JMS è stato installato e configurato correttamente in IBM MQ. È accessibile da un gestore code locale.

Per ulteriori informazioni, vedi [Client WCF a un servizio .NET ospitato dall'esempio IBM MQ](#) e [Client WCF a un servizio Axis Java ospitato dall'esempio IBM MQ](#).

XMS_TRACE_ON, XMS_TRACE_FILE_PATH, XMS_TRACE_FORMAT e XMS_TRACE_SPECIFICATION

Se si utilizza IBM MQ classes for XMS .NET Framework, è possibile configurare la traccia da un file di configurazione dell'applicazione e dalle variabili di ambiente XMS. Se si utilizza IBM MQ classes for XMS .NET (librerie.NET Standard e .NET 6), è necessario configurare la traccia dalle variabili di ambiente XMS. La traccia viene normalmente utilizzata sotto la guida del supporto IBM.

Per abilitare e configurare la traccia per un'applicazione XMS .NET, impostare le seguenti variabili di ambiente prima di eseguire l'applicazione:

TRACE_XMS_ON

Se la variabile di ambiente `XMS_TRACE_ON` è impostata, tutta la traccia è abilitata per impostazione predefinita.

PERCORSO_FILE_TRACCIA_X

La variabile di ambiente `XMS_TRACE_FILE_PATH` specifica il nome percorso completo della directory in cui vengono scritti i record FFDC e di traccia, se si desidera che tali record vengano scritti in un'ubicazione alternativa dalla directory di lavoro corrente.

FORMATO_TRACCIA_XMS

La variabile di ambiente `XMS_TRACE_FORMAT` specifica il formato di traccia richiesto, che può essere BASIC o ADVANCED.

SPECIFICA_TRACCIA_XMS

La variabile di ambiente **XMS_TRACE_SPECIFICATION** sostituisce le impostazioni di traccia definite nella sezione Traccia di un file di configurazione dell'applicazione. **XMS_TRACE_SPECIFICATION** si applica solo a IBM MQ classes for XMS .NET Framework .

Per ulteriori informazioni, vedi Traccia delle applicazioni XMS .NET e Traccia delle applicazioni XMS .NET utilizzando le XMS variabili di ambiente.

Multi **Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms**

È possibile modificare il comportamento di IBM MQ o di un singolo gestore code per adattarlo alle esigenze della propria installazione modificando le informazioni nei file di configurazione (.ini). È anche possibile modificare le opzioni di configurazione per IBM MQ MQI clients.

Informazioni su questa attività

È possibile modificare le informazioni di configurazione di IBM MQ a livello del nodo o del gestore code modificando i valori specificati in una serie di attributi di configurazione (o parametri) che gestiscono IBM MQ.


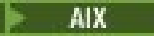
Un file di configurazione (o file di stanza) contiene una o più stanze, che sono gruppi di righe nel file .ini che insieme hanno una funzione comune o definiscono parte di un sistema, come funzioni di log, funzioni di canale e servizi installabili. È possibile modificare gli attributi di configurazione IBM MQ nei seguenti file di configurazione:

IBM MQ file di configurazione, mqs .ini

Il file mqs .ini influenza le modifiche sul nodo nel suo insieme. Esiste un file mqs .ini per ogni installazione di IBM MQ .

Poiché il file di configurazione IBM MQ viene utilizzato per individuare i dati associati ai gestori code, un file di configurazione non esistente o non corretto può causare l'esito negativo di alcuni o di tutti i comandi MQSC. Inoltre, le applicazioni non possono connettersi a un gestore code che non è definito nel file di configurazione IBM MQ .

File di configurazione dell'installazione, mqinst.ini

  Su sistemi AIX and Linux , il file di configurazione dell'installazione, mqinst .ini, contiene informazioni su tutte le installazioni di IBM MQ . Il file mqinst .ini non deve essere modificato o a cui si fa riferimento direttamente poiché il formato non è fisso e potrebbe essere modificato. Invece, è necessario modificarlo utilizzando i comandi.

File di configurazione del gestore code, qm .ini

Il file qm .ini influisce sulle modifiche per specifici gestori code. È presente un file qm .ini per ogni gestore code sul nodo.

IBM MQ MQI client file di configurazione, mqclient.ini

Le opzioni di configurazione per il IBM MQ MQI clients vengono conservate separatamente, nel file di configurazione del client, generalmente denominato mqclient .ini.

File di configurazione della traccia di attività, mqat.ini


Il file mqat .ini viene utilizzato per configurare il comportamento della traccia dell'attività.

Potrebbe essere necessario modificare un file di configurazione se, ad esempio:

- Si perde un file di configurazione. (Recuperarlo dal backup, se possibile.)
- È necessario spostare uno o più gestori code in una nuova directory.
- È necessario modificare il gestore code predefinito. Ciò può verificarsi se si elimina accidentalmente il gestore code esistente.
- Si consiglia di farlo dal supporto IBM .


Importante: Le modifiche apportate a un file di configurazione di solito non diventano effettive fino al successivo avvio del gestore code.



Punti da notare sulla modifica dei file di configurazione:

- I valori degli attributi di un file di configurazione sono impostati in base alla seguente priorità:
 - I parametri immessi sulla riga comandi hanno la precedenza sui valori definiti nei file di configurazione.
 - I valori definiti nei file `qm.ini` hanno la precedenza sui valori definiti nel file `mqs.ini`.
- Dopo l'installazione, è possibile modificare i valori predefiniti nei file di configurazione IBM MQ.
- Quando si esegue il backup di un gestore code, ricordarsi di includere sia il relativo file di configurazione (`qm.ini`) che il file di configurazione centrale IBM MQ (`mqs.ini`).
- Se si imposta un valore non corretto su un attributo del file di configurazione, l'effetto è lo stesso di perdere completamente l'attributo. Il valore viene ignorato e viene emesso un messaggio operatore per indicare il problema.
-  Su IBM i, i file `.ini` sono file di flusso residenti in IFS.
- Esistono diverse regole di sintassi per il formato del file `mqt.ini`. Per ulteriori informazioni, consultare la sezione Traccia attività dell'applicazione [Configurazione del comportamento della traccia attività utilizzando `mqt.ini`](#).

Procedura

1. Prima di modificare un file di configurazione, eseguirne il backup in modo da disporre di una copia a cui è possibile tornare, se necessario.
2. Modificare il file di configurazione `.ini` in uno dei seguenti modi:
 - Manualmente utilizzando un editor di testo standard. I commenti possono essere inclusi nei file di configurazione aggiungendo un carattere ";" o un carattere "#" prima del testo del commento. Se si desidera utilizzare un carattere ";" o un carattere "#" senza che rappresenti un commento, è possibile anteporre al carattere un carattere "\". Il carattere viene quindi utilizzato come parte dei dati di configurazione.
 - Automaticamente, utilizzando i comandi che modificano la configurazione dei gestori code sul nodo. Per ulteriori informazioni, consultare [Riferimento comandi](#).

 Ad esempio, il Windows comando specifico `amqmdain` aggiornerà automaticamente una serie secondaria delle proprietà `qm.ini`. Per ulteriori informazioni, consultare [amqmdain](#).

-   Su Linux (x86 e x86-64) e su Windows, è possibile aggiornare un sottoinsieme delle proprietà `qm.ini` utilizzando IBM MQ Explorer. Per ulteriori informazioni, fare riferimento a [Configurazione di IBM MQ utilizzando MQ Explorer](#).

Nota: Poiché ci sono implicazioni significative per la modifica dei servizi installabili e dei loro componenti, i servizi installabili sono di sola lettura in IBM MQ Explorer. Pertanto, è necessario apportare eventuali modifiche ai servizi installabili modificando il file `qm.ini`. Per ulteriori informazioni, consultare [“Stanza di servizio del file `qm.ini`”](#) a pagina 151.

Attività correlate

[Amministrazione IBM MQ](#)

 Multi

File di configurazione IBM MQ , `mqs.ini`

Il file di configurazione IBM MQ , `mqs.ini`, contiene informazioni relative a tutti i gestori code sul nodo. Viene creato automaticamente durante l'installazione.

Nota: Per ulteriori informazioni su come e quando modificare il file `mqs.ini` e su quando le modifiche apportate al file diventano effettive, consultare [“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms”](#) a pagina 93.

Percorsi di directory

Linux **AIX** Su AIX and Linux, la directory di dati e la directory di log sono sempre /var/mqm e /var/mqm/log rispettivamente.

Windows Sui sistemi Windows , l'ubicazione della directory di dati mqs . ini e l'ubicazione della directory di log sono memorizzate nel registro, poiché la loro ubicazione può variare. Le informazioni di configurazione dell'installazione, contenute in sistemi mqinst . ini su AIX and Linux , si trovano anche nel Registro di sistema, poiché non vi è alcun file mqinst . ini in Windows (consultare [“File di configurazione dell'installazione, mqinst.ini”](#) a pagina 167).

Windows Il file mqs . ini per i sistemi Windows viene fornito dal WorkPath specificato nella chiave HKLM\SOFTWARE\IBM\IBM MQ . Contiene:

- I nomi dei gestori code
- Il nome del gestore code predefinito
- L'ubicazione dei file associati a ciascuno di essi

IBM i Su IBM i, il file mqs . ini è memorizzato in /QIBM/UserData/mqm. Il file contiene:

- I nomi dei gestori code.
- Il nome del gestore code predefinito.
- L'ubicazione dei file associati a ciascun gestore code.
- Informazioni che identificano le uscite API (per ulteriori informazioni, consultare [Configurazione delle uscite API](#)).

In particolare, il file mqs . ini viene utilizzato per individuare i dati associati a ciascun gestore code.

File mqs . ini di esempio per AIX and Linux

Linux **AIX**

```
#####  
#* Module Name: mqs.ini                                     *#  
#* Type       : IBM MQ Machine-wide Configuration File     *#  
#* Function   : Define IBM MQ resources for an entire machine *#  
#####  
#* Notes      :                                           *#  
#* 1) This is the installation time default configuration *#  
#*                                                    *#  
#####  
AllQueueManagers:  
#* The path to the qmgrs directory, below which queue manager data *#  
#* is stored                                               *#  
#####  
DefaultPrefix=/var/mqm  
  
LogDefaults:  
  LogPrimaryFiles=3  
  LogSecondaryFiles=2  
  LogFilePages=4096  
  LogType=CIRCULAR  
  LogBufferPages=0  
  LogDefaultPath=/var/mqm/log  
  
QueueManager:  
  Name=saturn.queue.manager  
  Prefix=/var/mqm  
  Directory=saturn!queue!manager  
  InstallationName=Installation1  
  
QueueManager:  
  Name=pluto.queue.manager  
  Prefix=/var/mqm  
  Directory=pluto!queue!manager  
  InstallationName=Installation2
```

```

DefaultQueueManager:
  Name=saturn.queue.manager

ApiExitTemplate:
  Name=OurPayrollQueueAuditor
  Sequence=2
  Function=EntryPoint
  Module=/usr/ABC/auditor
  Data=123

ApiExitCommon:
  Name=MQPoliceman
  Sequence=1
  Function=EntryPoint
  Module=/usr/MQPolice/tmpq
  Data=CheckEverything

```

File mqs .ini di esempio per Windows

Windows

```

#*****#
#* Module Name: mqs.ini                                     **#
#* Type       : IBM MQ Machine-wide Configuration File    **#
#* Function   : Define IBM MQ resources for an entire machine **#
#*****#
#* Notes     :                                           **#
#* 1) This is the installation time default configuration **#
#*                                                  **#
#*****#
AllQueueManagers:
#*****#
#* The path to the qmgs directory, below which queue manager data **#
#* is stored                                                  **#
#*****#
DefaultPrefix=C:\ProgramData\IBM\MQ

LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0
  LogDefaultPath=C:\ProgramData\IBM\MQ\log

QueueManager:
  Name=saturn.queue.manager
  Prefix=C:\ProgramData\IBM\MQ
  Directory=saturn!queue!manager
  InstallationName=Installation1

QueueManager:
  Name=pluto.queue.manager
  Prefix=C:\ProgramData\IBM\MQ
  Directory=pluto!queue!manager
  InstallationName=Installation2

DefaultQueueManager:
  Name=saturn.queue.manager

ApiExitTemplate:
  Name=OurPayrollQueueAuditor
  Sequence=2
  Function=EntryPoint
  Module=C:\usr\ABC\auditor
  Data=123

ApiExitCommon:
  Name=MQPoliceman
  Sequence=1
  Function=EntryPoint
  Module=C:\usr\MQPolice\tmpq
  Data=CheckEverything

```


File mqs . ini di esempio per IBM i

IBM i

```

#*****#
#* Module Name: mqs.ini                               *#
#* Type       : IBM MQ Configuration File             *#
#* Function   : Define IBM MQ resources for the node  *#
#*           *#
#*****#
#* Notes     :                                       *#
#* 1) This is an example IBM MQ configuration file    *#
#*           *#
#*****#
AllQueueManagers:
#*****#
#* The path to the qmgrs directory, within which queue manager data *#
#* is stored                                           *#
#*****#
DefaultPrefix=/QIBM/UserData/mqm

QueueManager:
Name=saturn.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMSATURN.Q
Directory=saturn!queue!manager

QueueManager:
Name=pluto.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMPLUTO.QU
Directory=pluto!queue!manager

DefaultQueueManager:
Name=saturn.queue.manager

```

Note:

1. IBM MQ sul nodo utilizza le ubicazioni predefinite per i gestori code e i journal.
2. Il gestore code saturn.queue.manager è il gestore code predefinito per il nodo. La directory per i file associati a questo gestore code è stata automaticamente trasformata in un nome file valido per il filesystem.
3. Poiché il file di configurazione IBM MQ viene utilizzato per individuare i dati associati ai gestori code, un file di configurazione non esistente o non corretto può causare l'esito negativo di alcuni o di tutti i comandi IBM MQ . Inoltre, le applicazioni non possono connettersi a un gestore code che non è definito nel file di configurazione IBM MQ .

stanze mqs . ini



Attenzione: Questo argomento si collega a ulteriori informazioni sulle stanze nel file mqs . ini . Ogni stanza contiene informazioni sui parametri in quella stanza.

Multi

Riepilogo delle stanze e degli attributi del file mqs.ini

Un riepilogo degli attributi delle stanze del IBM MQ file di configurazione, mqs . ini, con link a ulteriori informazioni.


Tabella 10. Stanze del file mqs.ini	
Stanza e attributi	Descrizione degli attributi
Stanza AllQueueManagers	
DefaultPrefix	Il percorso della directory qmgrs , all'interno della quale vengono conservati i dati del gestore code.
 DefaultEphemeralpredefinito	Il percorso della directory in cui vengono conservati i dati effimeri del gestore code.

Tabella 10. Stanze del file mqs.ini (Continua)


Stanza e attributi	Descrizione degli attributi
 <u>ConvEBCDICNewline</u>	Come IBM MQ deve convertire il carattere EBCDIC NL in formato ASCII
ApiExitStanza comune e stanza ApiExitTemplate	
<u>Nome</u>	Il nome descrittivo dell'uscita API inoltrato nel campo Nome ExitInfodella struttura MQAXP.
<u>funzione</u>	Il nome del punto di ingresso della funzione nel modulo contenente il codice di uscita API.
<u>Modulo</u>	Il modulo contenente il codice di uscita API.
<u>Dati</u>	I dati da passare all'uscita API nel campo ExitData della struttura MQAXP.
<u>Sequenza</u>	La sequenza in cui questa uscita API viene richiamata rispetto ad altre uscite API.
Stanza del gestoreDefaultQueue	
<u>Nome</u>	Il nome del gestore code che elabora i comandi per cui un nome gestore code non è specificato esplicitamente.
Stanza ExitProperties	
<u>CLWLMode</u>	Indica se l'uscita CLWL (cluster workloac) viene eseguita in modalità FAST o SAFE.
Stanza LogDefaults	
<u>LogPrimaryFiles</u>	I file di log assegnati quando viene creato il gestore code.
<u>LogSecondaryFiles</u>	I file di log assegnati quando i file primari sono esauriti.
<u>LogFilePages</u>	Il numero di pagine del file di log. (La dimensione del file di log è specificata in unità di pagine da 4 KB.)
<u>LogType</u>	Il tipo di registrazione che deve essere utilizzato dal gestore code (circolare o lineare).
<u>LogBufferPages</u>	La quantità di memoria assegnata ai record buffer per la scrittura, specificando la dimensione dei buffer in unità di pagine da 4 KB.
<u>LogDefaultPath</u>	La directory in cui risiedono i file di log per un gestore code.
<u>LogWriteIntegrity</u>	Il metodo utilizzato dal programma di registrazione per scrivere in modo affidabile i record di log.
StanzaQueueManager	
<u>Nome</u>	Il nome del gestore code.
<u>prefisso</u>	La posizione in cui sono memorizzati i file del gestore code.
<u>Directory</u>	Il nome della sottodirectory nella directory prefix\QMGRS in cui sono memorizzati i file del gestore code.

Tabella 10. Stanze del file `mqs.ini` (Continua)

Stanza e attributi	Descrizione degli attributi
DataPath	Un percorso dati esplicito fornito quando è stato creato il gestore code, sostituisce Prefisso e Directory come percorso dei dati del gestore code.
InstallationName	Il nome dell'installazione IBM MQ associato a questo gestore code.
EphemeralPrefix	Dove vengono memorizzati i dati effimeri del gestore code.

Multi Stanza AllQueueManagers del file `mqs.ini`

La stanza AllQueueManagers può specificare il percorso della directory `qmgrs` in cui sono memorizzati i file associati a un gestore code, il percorso della libreria eseguibile e il metodo di conversione dei dati in formato EBCDIC in formato ASCII.

Utilizzare la stanza AllQueueManagers nel file `mqs.ini` per specificare le informazioni su tutti i gestori code.

Linux **Windows** In alternativa, in Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà IBM MQ Explorer General e Extended IBM MQ .

DefaultPrefix= nome_directory

Questo attributo specifica il percorso della directory `qmgrs` , in cui vengono conservati i dati del gestore code.

Se si modifica il prefisso predefinito per il gestore code, replicare la struttura di directory creata al momento dell'installazione. In particolare, è necessario creare la struttura `qmgrs`. Arrestare IBM MQ prima di modificare il prefisso predefinito e riavviare IBM MQ solo dopo aver spostato le strutture nella nuova posizione e aver modificato il prefisso predefinito.

Nota: **ALW** Non eliminare la directory `/var/mqm/errors` su sistemi AIX and Linux o la directory `\errors` su sistemi Windows .

In alternativa alla modifica del prefisso predefinito, è possibile utilizzare la variabile di ambiente **MQSPREFIX** per sovrascrivere **DefaultPrefix** per il comando **crtmqm** .

A causa delle limitazioni del sistema operativo, mantenere il percorso fornito sufficientemente breve in modo che la somma della lunghezza del percorso e dei nomi dei gestori code abbia una lunghezza massima di 70 caratteri.

Multi DefaultEphemeralPrefisso = nome_directory

Questo attributo specifica il percorso della directory, all'interno della quale vengono conservati i dati temporanei del gestore code, come i socket IPC, e viene utilizzato solo per impostare il **EphemeralPrefix** di un gestore code quando viene creato un gestore code. Inoltre, è necessario creare la directory se si modifica il valore predefinito. È necessario creare la directory di dati effimeri con le autorizzazioni che consentono al gruppo IBM MQ di accedere alla scrittura in tale directory.

Come alternativa alla modifica del file `mqs.ini` , è possibile utilizzare la variabile di ambiente **MQ_EPHEMERAL_PREFIX** per sovrascrivere il **DefaultEphemeralPrefix** per il comando **crtmqm** .

A causa delle limitazioni del sistema operativo, il prefisso effimero predefinito è limitato a:

- **Linux** **AIX** 12 caratteri su piattaforme AIX and Linux .
- **IBM i** 24 caratteri su IBM i.

MQ Appliance **DefaultEphemeralPrefix** non è supportato su IBM MQ Appliance.

Multi ConvEBCDICNewline= NL_TO_LF | TABLE | ISO

Le codepage EBCDIC contengono un carattere di nuova riga (NL) che non è supportato dalle codepage ASCII (anche se alcune varianti ISO di ASCII contengono un equivalente). Utilizzare l'attributo **ConvEBCDICNewline** per specificare come IBM MQ deve convertire il carattere EBCDIC NL in formato ASCII.

IBM i Su IBM MQ for IBM i, CCSID 1253 viene considerato come un CCSID ISO e NL_TO_LF influisce sulle conversioni ISO e ASCII.

z/OS L'attributo **ConvEBCDICNewline** non è disponibile su z/OS. Il comportamento su z/OS equivale a ConvEBCDICNewline=TABLE. Notare che il valore predefinito su altre piattaforme potrebbe essere diverso.

NL_TO_LF

Convertire il carattere NL EBCDIC (X'15 ') nel carattere di avanzamento riga ASCII, LF (X'0A'), per tutte le conversioni da EBCDIC a ASCII.

NL_TO_LF è il valore predefinito.

TABELLA

Convertire il carattere EBCDIC NL in base alle tabelle di conversione utilizzate sulla piattaforma per tutte le conversioni da EBCDIC ad ASCII.

L'effetto di questo tipo di conversione può variare da piattaforma a piattaforma e da lingua a lingua; anche sulla stessa piattaforma, il comportamento potrebbe variare se si utilizzano CCSID differenti.

ISO

Converti:

- CCSID ISO che utilizzano il metodo TABLE
- Tutti gli altri CCSID che utilizzano il metodo NL_TO_LF

I CCSID ISO possibili sono mostrati in [Tabella 11 a pagina 100](#).

CCSID	Serie di codici
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

Se il CCSID ASCII non è un sottoinsieme ISO, **ConvEBCDICNewline** assume il valore predefinito NL_TO_LF.

Da IBM MQ 9.1.0 Fix Pack 2 e IBM MQ 9.1.2, è possibile utilizzare la variabile di ambiente **AMQ_CONVEBDICNEWLINE** invece dell'attributo stanza **ConvEBCDICNewline**, ad esempio per fornire la funzionalità **ConvEBCDICNewline** sul lato client in situazioni in cui non è possibile utilizzare il file `mqs.ini`. La variabile di ambiente assume gli stessi valori (NL_TO_LF, TABLE o ISO) dell'attributo **ConvEBCDICNewline**. L'attributo stanza ha la precedenza se sono impostati sia l'attributo che la variabile di ambiente.

Stanze del modello ApiExitCommon e ApiExitdel file mqs.ini

Il modello ApiExite le stanze comuni ApiExitidentificano le routine di uscita API per tutti i gestori code.

Utilizzare il modello ApiExite le stanze comuni ApiExitnel file mqs . ini per identificare le routine di uscita API per tutti i gestori code. (Per identificare le routine di uscita API per singoli gestori code, utilizzare la stanza ApiExitlocale, come descritto in [“ApiExitStanza locale del file qm.ini”](#) a pagina 119.)

In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà di IBM MQ Explorer Exits IBM MQ .

Su Windows, è anche possibile utilizzare il comando **amqmdain** per modificare le voci per le uscite API.

Per ulteriori informazioni sull'utilizzo di questi attributi, consultare [Configurazione delle uscite API](#).

Nome=ApiExit_name

Il nome descrittivo dell'uscita API inoltrato nel campo Nome ExitInfodella struttura MQAXP.

Questo nome deve essere univoco, non deve superare i 48 caratteri e contenere solo caratteri validi per i nomi degli oggetti IBM MQ (ad esempio, nomi coda).

Funzione=nome_funzione

Il nome del punto di ingresso della funzione nel modulo contenente il codice di uscita API. Questo punto di ingresso è la funzione MQ_INIT_EXIT.

La lunghezza di questo campo è limitata a MQ_EXIT_NAME_LENGTH.

Modulo=nome_modulo

Il modulo contenente il codice di uscita API.

Se questo campo contiene il percorso completo del modulo, questo verrà visualizzato così come è. Se questo campo contiene solo il nome del modulo, il modulo si trova utilizzando l'attributo

ExitsDefaultPath nella stanza ExitPath del file qm . ini .

Su piattaforme che supportano librerie separate, è necessario fornire sia una versione non con thread che una versione con thread del modulo di uscita API. La versione con thread deve avere un suffisso **_t** . La versione con thread dello stub dell'applicazione IBM MQ accoda implicitamente **_t** al nome modulo fornito prima che venga caricato.

La lunghezza di questo campo è limitata alla lunghezza massima del percorso supportata dalla piattaforma.

Dati=nome_dati

I dati da passare all'uscita API nel campo ExitData della struttura MQAXP.

Se si include questo attributo, vengono rimossi gli spazi iniziali e finali, la stringa rimanente viene troncata a 32 caratteri e il risultato viene passato all'uscita. Se si omette questo attributo, il valore predefinito di 32 spazi viene passato all'uscita.

La lunghezza massima di questo campo è 32 caratteri.

Sequenza=numero_sequenza

La sequenza in cui questa uscita API viene richiamata rispetto ad altre uscite API. Un'uscita con un numero di sequenza basso viene richiamata prima di un'uscita con un numero di sequenza più alto. Non è necessario che la numerazione di sequenza delle uscite sia contigua. Una sequenza di 1, 2, 3 ha lo stesso risultato di una sequenza di 7, 42, 1096. Se due uscite hanno lo stesso numero di sequenza, il gestore code decide quale richiamare per primo. È possibile determinare quale elemento è stato richiamato dopo l'evento inserendo l'ora o un indicatore nell'area ExitChainindicata da ExitChainAreaPtr in MQAXP oppure scrivendo il proprio file di log.

Questo attributo è un valore numerico senza segno.

Stanza del gestore DefaultQueuedel file mqs.ini

La stanza DefaultQueueManager specifica il gestore code predefinito per il nodo.

Utilizzare la stanza Gestore DefaultQueuenel file mqs . ini per specificare il gestore code predefinito.

Linux **Windows** In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la IBM MQ Explorer pagina delle proprietà General IBM MQ .

Nome = *default_queue_manager*

Il gestore code predefinito elabora tutti i comandi per cui non è specificato esplicitamente un nome gestore code. L'attributo **DefaultQueueManager** viene aggiornato automaticamente se si crea un nuovo gestore code predefinito. Se si crea inavvertitamente un nuovo gestore code predefinito e si desidera ripristinare l'originale, modificare manualmente l'attributo **DefaultQueueManager** .

Multi Stanza ExitProperties del file mqs.ini

La stanza ExitProperties specifica le opzioni di configurazione utilizzate dai programmi di uscita del gestore code.

Utilizzare la stanza ExitProperties nel file mqs . ini per specificare opzioni di configurazione utilizzate dai programmi di uscita del gestore code.

Linux **Windows** In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà di IBM MQ Explorer Extended IBM MQ .

CLWLMode = **SAFE (predefinito) | FAST**

L'uscita del carico di lavoro cluster (CLWL) consente di specificare quale coda cluster nel cluster aprire in risposta a una chiamata MQI (ad esempio, MQOPEN, MQPUT). L'uscita CLWL viene eseguita in modalità FAST o SAFE in base al valore specificato nell'attributo **CLWLMode** . Se si omette l'attributo **CLWLMode** , l'uscita del carico di lavoro cluster viene eseguita in modalità SAFE.

SICURA

Eseguire l'uscita CLWL in un processo separato dal gestore code. Questa è l'opzione predefinita.

Se si verifica un problema con l'uscita CLWL scritta dall'utente durante l'esecuzione in modalità SAFE, si verifica quanto segue:

- Il processo del server CLWL (amqzlw0) ha esito negativo.
- Il gestore code riavvia il processo server CLWL.
- L'errore viene riportato nel log degli errori. Se è in corso una chiamata MQI, si riceve una notifica sotto forma di codice di ritorno.

L'integrità del gestore code viene preservata.

Nota: L'esecuzione dell'uscita CLWL in un processo separato potrebbe influire sulle prestazioni.

VELOCE

Eseguire l'uscita cluster in linea nel processo del gestore code.

Specificando questa opzione si migliorano le prestazioni evitando i costi di commutazione del processo associati all'esecuzione in modalità SAFE, ma a discapito dell'integrità del gestore code. Si consiglia di eseguire l'uscita CLWL in modalità FAST solo se si è certi che non vi sono problemi con l'uscita CLWL e si è particolarmente preoccupati per le prestazioni.

Se si verifica un problema quando l'uscita CLWL è in esecuzione in modalità FAST, il gestore code avrà esito negativo e si corre il rischio che l'integrità del gestore code venga compromessa.

Multi Stanza LogDefaults del file mqs.ini

La stanza LogDefaults specifica le informazioni sui valori predefiniti di log per tutti i gestori code.

Utilizzare la stanza LogDefaults nel file mqs . ini per specificare le informazioni sui valori predefiniti di log per tutti i gestori code.

Linux **Windows** In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà di IBM MQ Explorer Default log settings IBM MQ .

Se si richiede un valore non predefinito, è necessario specificare esplicitamente tale valore nella stanza LogDefaults .

Se la stanza LogDefaults non esiste, vengono utilizzati i valori predefiniti IBM MQ . Gli attributi di log vengono utilizzati come valori predefiniti quando si crea un gestore code, ma possono essere sovrascritti se si specificano gli attributi di log nel comando **crtmqm** . Per ulteriori informazioni su questo comando, consultare **crtmqm**.

Una volta creato un gestore code, gli attributi di log per tale gestore code vengono ricavati dalle impostazioni descritte in “Stanza di log del file qm.ini” a pagina 141.

Nota: La stanza LogDefaults fornita per una nuova installazione IBM MQ non contiene alcun valore esplicito per gli attributi. La mancanza di un attributo indica che l'impostazione predefinita per questo valore viene utilizzata al momento della creazione di un nuovo gestore code. I valori predefiniti per la stanza LogDefaults sono riportati in “File mqs.ini di esempio per AIX and Linux” a pagina 95 e “File mqs.ini di esempio per Windows” a pagina 96. Un valore zero per l'attributo LogBufferPages indica 512.

Il prefisso predefinito, specificato in “Stanza AllQueueManagers del file mqs.ini” a pagina 99, e il percorso di log specificato per il particolare gestore code, specificato in “Stanza di log del file qm.ini” a pagina 141, consentono al gestore code e al relativo log di trovarsi su unità fisiche differenti. Questo è il metodo consigliato, anche se per impostazione predefinita si trovano sulla stessa unità.

Per informazioni sul calcolo delle dimensioni del log, consultare “Calcolo della dimensione del log” a pagina 671.

Nota: I limiti forniti nel seguente elenco di parametri sono limiti impostati da IBM MQ. I limiti del sistema operativo potrebbero ridurre la dimensione massima possibile del log.

LogPrimaryFiles = 3 (predefinito) | 2-254 (Windows) | 2 - 510 (AIX and Linux)

I file di log assegnati quando viene creato il gestore code.

Il numero minimo di file di log primari che è possibile avere è 2 e il numero massimo è 254 su Windows e 510 su AIX and Linux. Il valore predefinito è 3.

Il numero totale di file di log primari e secondari non deve superare 255 su Windows e 511 su AIX and Linux e non deve essere inferiore a 3.

Una volta creato o avviato il gestore code, il valore viene configurato automaticamente. È possibile modificarlo una volta creato il gestore code. Tuttavia, una modifica del valore non è effettiva fino a quando il gestore code non viene riavviato e l'effetto potrebbe non essere immediato.

LogSecondaryFiles = 2 (predefinito) | 1-253 (Windows) | 1-509 (AIX and Linux)

I file di log assegnati quando i file primari sono esauriti.

Il numero minimo di file di log secondari è 1 e il numero massimo è 253 su Windows e 509 su AIX and Linux. Il numero predefinito è 2.

Il numero totale di file di log primari e secondari non deve superare 255 su Windows e 511 su AIX and Linux e non deve essere inferiore a 3.

Il valore viene esaminato quando il gestore code viene avviato. È possibile modificare questo valore, ma le modifiche non diventano effettive fino a quando il gestore code non viene riavviato e anche in questo caso l'effetto potrebbe non essere immediato.

LogFilePage = numero

I dati di log sono contenuti in una serie di file denominati file di log. La dimensione del file di log è specificata in unità di pagine da 4 KB.

Il numero predefinito di pagine del file di log è 4096, fornendo una dimensione del file di log di 16 MB.

Su AIX and Linux, il numero minimo di pagine del file di log è 64, e su Windows, il numero minimo di pagine del file di log è 32; in entrambi i casi, il numero massimo è 65 535.

Nota: La dimensione dei file di log specificati durante la creazione del gestore code non può essere modificata per un gestore code.

LogType = CIRCULAR (valore predefinito) | LINEAR

Il tipo di log da utilizzare. Il valore predefinito è CIRCULAR.

CIRCOLARE

Avviare il ripristino utilizzando il log per eseguire il rollback delle transazioni che erano in corso quando il sistema è stato arrestato.

Consultare [“Tipi di registrazione” a pagina 665](#) per una spiegazione più completa della registrazione circolare.

LINEARE

Sia per il ripristino del riavvio che per il ripristino del supporto o dell'inoltro (creazione di dati persi o danneggiati riproducendo il contenuto della registrazione).

Consultare [“Tipi di registrazione” a pagina 665](#) per una spiegazione più completa della registrazione lineare.

Se si desidera modificare il valore predefinito, è possibile modificare l'attributo LogType oppure specificare la registrazione lineare utilizzando il comando **crtmqm**.

È possibile modificare il metodo di registrazione dopo che è stato creato un gestore code. Per ulteriori informazioni, consultare [migmqlog](#).

LogBufferPages=0 (predefinito) | 0 - 4096

La quantità di memoria assegnata ai record buffer per la scrittura, specificando la dimensione dei buffer in unità di pagine da 4 KB.

Il numero minimo di pagine di buffer è 18 e il massimo è 4096. Buffer più grandi portano ad una maggiore velocità di trasmissione, specialmente per messaggi più grandi.




Se si specifica 0 (valore predefinito), il gestore code seleziona la dimensione 512 (2048 KB).

Se si specifica un numero compreso tra 1 e 17, il valore predefinito del gestore code è 18 (72 KB). Se si specifica un numero compreso nell'intervallo tra 18 e 4096, il gestore code utilizza il numero specificato per impostare la memoria assegnata.

LogDefaultPath = nome_directory

La directory in cui risiedono i file di log per un gestore code. La directory risiede su una periferica locale in cui il gestore code può scrivere e, preferibilmente, su un'unità differente dalle code di messaggi. La specifica di un'unità differente fornisce una protezione aggiuntiva in caso di errore del sistema.

Il valore predefinito è:

-  *DefaultPrefix*\log per IBM MQ for Windows dove *DefaultPrefix* è il valore specificato nell'attributo DefaultPrefix nella pagina delle proprietà di All Queue Managers IBM MQ. Questo valore è impostato al momento dell'installazione.
-   /var/mqm/log per sistemi AIX and Linux.

In alternativa, è possibile specificare il nome di una directory nel comando **crtmqm** utilizzando l'indicatore **-ld**. Quando un gestore code viene creato, viene creata anche una directory nella directory del gestore code, utilizzata per conservare i file di log. Il nome di questa directory è basato sul nome gestore code. Ciò garantisce che il percorso del file di log sia univoco e che sia conforme alle eventuali limitazioni sulle lunghezze dei nomi di directory.

Se non si specifica **-ld** nel comando **crtmqm**, viene utilizzato il valore dell'attributo **LogDefaultPath** nel file `mqs.ini`.

Il nome del gestore code viene aggiunto al nome della directory per garantire che più gestori code utilizzino directory di log differenti.

Quando il gestore code viene creato, viene creato un valore **LogPath** negli attributi di log nelle informazioni di configurazione, fornendo il nome completo della directory per il log del gestore code. Questo valore viene utilizzato per individuare il log quando il gestore code viene avviato o eliminato.

LogWriteIntegrity =SingleWrite|DoubleWrite|TripleWrite (predefinito)

Il metodo utilizzato dal programma di registrazione per scrivere in modo affidabile i record di log.

TripleWrite (predefinito)

Nota: è possibile selezionare DoubleWrite ma, in tal caso, il sistema l'interpreta come TripleWrite.

SingleWrite

Si consiglia di utilizzare SingleWrite, solo se il file - system e il dispositivo che ospita il log di recupero IBM MQ garantiscono esplicitamente l'atomicità delle scritture 4KB .

Ossia, quando una scrittura di una pagina di 4KB non riesce per un qualsiasi motivo, i soli due stati possibili sono la pre-immagine o la post-immagine. Non deve essere possibile alcuno stato intermedio.

Nota: Se è presente una simultaneità sufficiente nel tuo carico di lavoro persistente, c'è un vantaggio potenziale minimo nell'impostazione di un valore diverso da quello predefinito, TripleWrite.

Per ulteriori informazioni, consultare [“LogWriteIntegrity - utilizzando SingleWrite o TripleWrite” a pagina 144.](#)

Multi

Stanza QueueManager del file mqs.ini

La stanza QueueManager specifica l'ubicazione della directory del gestore code.

Esiste una stanza QueueManager per ogni gestore code. Gli attributi di questa stanza specificano il nome del gestore code e il nome della directory contenente i file associati a tale gestore code. Il nome della directory si basa sul nome del gestore code, ma viene trasformato se il nome del gestore code non è un nome file valido. Per ulteriori informazioni sulla trasformazione dei nomi, consultare [Informazioni sui nomi file IBM MQ](#).

Nome = nome_gestore_coda

Il nome del gestore code.

Prefisso = prefisso

La posizione in cui sono memorizzati i file del gestore code. Per impostazione predefinita, questo valore è uguale al valore specificato nell'attributo **DefaultPrefix** della sezione [Tutti i gestori code](#) nel file mqs.ini .

Directory = nome

Il nome della sottodirectory nella directory *prefix*\QMGRS in cui sono memorizzati i file del gestore code. Questo nome si basa sul nome del gestore code, ma può essere trasformato se è presente un nome duplicato o se il nome del gestore code non è un nome file valido.

DataPath= percorso

Un percorso dati esplicito fornito al momento della creazione del gestore code, sovrascrive **Prefix** e **Directory** come percorso dei dati del gestore code.

InstallationName= nome

Il nome dell'installazione IBM MQ associato a questo gestore code. I comandi da questa installazione devono essere utilizzati quando si interagisce con questo gestore code.

IBM i

Libreria = nome

Il nome della libreria in cui sono memorizzati gli oggetti IBM i pertinenti a questo gestore code, ad esempio i giornali e i ricevitori di journal. Questo nome si basa sul nome del gestore code, ma può essere trasformato se è presente un nome duplicato o se il nome del gestore code non è un nome libreria valido.

EphemeralPrefix= nome

Dove vengono memorizzati i dati effimeri del gestore code.

Per impostazione predefinita, questo valore non è presente, il che significa che i dati vengono memorizzati nell'ubicazione Prefisso.

Il valore viene impostato dal valore della variabile di ambiente **MQ_EPHEMERAL_PREFIX** o dall'attributo **DefaultEphemeralPrefix** della stanza AllQueueManagers nel file `mqs.ini`, quando il gestore code viene creato.

IBM i A causa delle limitazioni del sistema operativo, il prefisso temporaneo predefinito è limitato a 24 caratteri su IBM i.

Attività correlate

“Associazione di un gestore code a un'installazione” a pagina 481

Quando si crea un gestore code, viene automaticamente associato all'installazione che ha emesso il comando **crtmqm**. Su AIX, Linux, and Windows, è possibile modificare l'installazione associata a un gestore code utilizzando il comando **setmqm**.

Windows API (Advanced Configuration and Power Interface)

Windows supporta lo standard ACPI (Advanced Configuration and Power Interface). Ciò consente agli utenti Windows con hardware abilitato ACPI di arrestare e riavviare i canali quando il sistema entra e riprende dalla modalità di sospensione.

Utilizzare la pagina delle proprietà di ACPI IBM MQ da IBM MQ Explorer per specificare come IBM MQ deve comportarsi quando il sistema riceve una richiesta di sospensione.

Tenere presente che le impostazioni specificate nella pagina delle proprietà di ACPI IBM MQ vengono applicate solo quando il Controllo segnalazioni è in esecuzione. L'icona Controllo segnalazioni è presente sulla barra delle attività se il Controllo segnalazioni è in esecuzione.

DoDialog= Y | N

Visualizza la finestra di dialogo al momento della richiesta di sospensione.

DenySuspend= Y | N

Nega la richiesta di sospensione. Questa opzione viene utilizzata se DoDialog= N o se DoDialog= Y e una finestra di dialogo non possono essere visualizzati, ad esempio, perché il coperchio del notebook è chiuso.

CheckChannelsin esecuzione=Y | N

Verifica se i canali sono in esecuzione. Il risultato può determinare il risultato delle altre impostazioni.

La seguente tabella illustra l'effetto di ciascuna combinazione di questi parametri:

DoDialog	DenySuspend	CheckChannels in esecuzione	Azione
N	N	N	Accettare la richiesta di sospensione.
N	N	Y	Accettare la richiesta di sospensione.
N	Y	N	Negare la richiesta di sospensione.
N	Y	Y	Se sono in esecuzione canali, negare la richiesta di sospensione; in caso contrario, accettare la richiesta.
Y	N	N	Visualizzare la finestra di dialogo (vedere Nota ; accettare la richiesta di sospensione). Questa è l'opzione predefinita.
Y	N	Y	Se nessun canale è in esecuzione, accettare la richiesta di sospensione; se viene visualizzata la finestra di dialogo (vedere Nota ; accettare la richiesta).
Y	Y	N	Visualizzare la finestra di dialogo (Nota ; negare la richiesta di sospensione).

Y	Y	Y	Se non è in esecuzione alcun canale, accettare la richiesta di sospensione; se viene visualizzata la finestra di dialogo (Nota ; negare la richiesta).
---	---	---	---

Nota: Nei casi in cui l'azione consiste nel visualizzare la finestra di dialogo, se la finestra di dialogo non può essere visualizzata (ad esempio perché il coperchio del notebook è chiuso), l'opzione DenySuspend viene utilizzata per stabilire se la richiesta di sospensione viene accettata o negata.

Multi File di configurazione del gestore code, qm.ini

Un file di configurazione del gestore code, `qm.ini`, contiene informazioni relative a uno specifico gestore code. Gli attributi che è possibile utilizzare per modificare la configurazione di un singolo gestore code sovrascrivono le impostazioni per IBM MQ.

Esiste un file di configurazione del gestore code per ciascun gestore code. Il file `qm.ini` viene creato automaticamente quando viene creato il gestore code a cui è associato.

Nota: Per ulteriori informazioni su come e quando modificare un file `qm.ini` e su quando le modifiche apportate al file diventano effettive, consultare [“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms”](#) a pagina 93.

Il comando `strmqm` controlla la sintassi delle stanze CHANNELS e SSL nel file `qm.ini` prima di avviare completamente il gestore code, il che rende molto più semplice visualizzare l'errore e correggerlo rapidamente se `strmqm` rileva che il file `qm.ini` contiene errori. Per ulteriori informazioni, vedere [strmqm](#).

Ubicazione dei file qm.ini

Linux **AIX** Sui sistemi AIX and Linux , un file di `qm.ini` è contenuto nella root della struttura di directory occupata dal gestore code. Ad esempio, il percorso e il nome di un file di configurazione per un gestore code denominato QMNAME è:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows Su sistemi Windows , l'ubicazione del file `qm.ini` viene fornita dal WorkPath specificato nella chiave HKLM\SOFTWARE\IBM\WebSphere MQ . Ad esempio, il percorso e il nome per un file di configurazione per un gestore code denominato QMNAME è il seguente:

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

IBM i Un file `qm.ini` viene conservato in `mqmdata directory/QMNAME/qm.ini`, dove `mqmdata directory` è `/QIBM/UserData/mqm` per impostazione predefinita e `QMNAME` è il nome del gestore code a cui si applica il file di inizializzazione.

Nota: È possibile modificare `mqmdata directory` nel file `mq.s.ini` .

Il nome del gestore code può avere una lunghezza massima di 48 caratteri. Tuttavia, ciò non garantisce che il nome sia valido o univoco. Pertanto, viene generato un nome di directory basato sul nome del gestore code. Questo processo è noto come *trasformazione del nome*. Per una descrizione, consultare [IBM MQ file names e Object names on IBM i](#).

stanze qm.ini



Attenzione:

- Questo argomento si collega a ulteriori informazioni sulle stanze nel file `qm.ini`. Ogni stanza contiene informazioni sui parametri presenti in quella stanza, incluso un esempio dove appropriato.
- Ogni stanza mostra la piattaforma, o le piattaforme, di IBM MQ for Multiplatforms a cui si applica tale stanza.

Multi

Configurazione automatica di `qm.ini` all'avvio

È possibile configurare il gestore code in modo che applichi automaticamente il contenuto di un file o di una serie di file, contenenti le sovrascritture `qm.ini`, ad ogni avvio del gestore code.

È possibile utilizzarlo per avere una configurazione che può essere modificata e riprodotta automaticamente al successivo riavvio del gestore code. Ad esempio, se le sovrascritture di `qm.ini` 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.

È possibile utilizzare questa funzionalità per semplificare la creazione di un cluster uniforme, utilizzando la funzionalità cluster automatica. Per un esempio, consultare [“Creazione di un nuovo cluster uniforme”](#) a pagina 442.

Nota: Queste sovrascritture vengono applicate solo all'avvio del gestore code e non possono influenzare la creazione del gestore code. Ad esempio, non è possibile impostare il numero di file di log primari con questa funzione.

Prima di iniziare

È possibile utilizzare:

1. Un singolo file e creare un file di testo contenente le modifiche al file `qm.ini`.
2. Un insieme di file di formato `qm.ini`:
 - Per identificare una directory in cui esisteranno le configurazioni e
 - In tale directory, creare i file, ognuno con estensione `.ini`, ad esempio `qminisettings.ini`.

Il file, o i file, devono contenere solo la sezione e le impostazioni **attribute=value** per gli elementi che vengono modificati. Ad esempio, per aggiornare l'attributo **MaxChannels** nella stanza Channels, il file potrebbe contenere:

```
Channels:
MaxChannels=1234
```

Tenere presente che nei file di sovrascrittura `qm.ini`, qualsiasi riga con prefisso `#` viene considerata come un commento.

Abilitazione della configurazione automatica degli attributi del file `qm.ini`

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

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

```
AutoConfig:
IniConfig=C:\MQ_Configuration\uniclus.ini
```

Come funziona la configurazione automatica?

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

Quando vengono elaborati più file da una directory, vengono elaborati in ordine alfabetico.

Durante il primo avvio del gestore code, l'impossibilità di leggere il file o la directory impedisce l'avvio del gestore code, con un messaggio di errore appropriato sia per la console che per il log degli errori del gestore code.

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

Quando si utilizza il comando **strmqm**, il contenuto del file `cached.ini` viene applicato al file `qm.ini` come sovrascrittura prima che venga richiamato il gestore code.

Ciò significa che per un gestore code in standby, le impostazioni vengono lette quando il comando **strmqm** viene elaborato, non quando il gestore code diventa attivo.

Come viene creato il file di sostituzione qm.ini ?

La prima volta che viene configurata la configurazione dell'inizializzazione automatica e il gestore code viene avviato, una copia del file `qm.ini` corrente viene copiata nella sottodirectory `autoconfig` all'interno della directory di dati del gestore code come `base_qm.ini`. Questa è considerata la linea di base da qui in poi.

Ad ogni avvio del gestore code, ossia all'ora **strmqm**, il file `qm.ini` attualmente attivo viene eliminato e sostituito con una copia di `base_qm.ini`. Quindi, la configurazione dal file `cached.ini` viene applicata a questo file.

Una volta che un gestore code è sotto il controllo della configurazione automatica, tutte le modifiche al file `qm.ini` devono essere eseguite tramite il file o i file a cui si fa riferimento utilizzando l'attributo **IniConfig** nella stanza **AutoConfig**.

Poiché il file `qm.ini` esistente viene rimosso all'avvio del gestore code, solo la configurazione nel file `qm.ini` fornito utilizzando l'attributo **IniConfig** viene applicata alla riga di base del gestore code.

Se una stanza o un attributo sono stati modificati tramite la configurazione di inizializzazione automatica nei precedenti avvii del gestore code, tali modifiche vengono rimosse a meno che non siano ancora identificate nel file o nei file identificati dall'attributo **IniConfig**.

A causa della ricreazione del file `qm.ini` all'avvio del gestore code, tutte le modifiche manuali al file `qm.ini` andranno perse. Se è necessario apportare una modifica persistente e non è possibile utilizzare l'attributo **IniConfig** per apportare tale modifica, è possibile effettuare una delle seguenti operazioni:

- Apportare la modifica al file `base_qm.ini`.
- Eliminare il file `base_qm.ini`.

Se si elimina questo file, `base_qm.ini` viene ricreato al successivo avvio del gestore code, in base al contenuto corrente del file `qm.ini`. Questo *rarisce* tutte le modifiche correnti man mano che inizia la nuova baseline per il futuro.

Concetti correlati

[“Riepilogo delle stanze e degli attributi del file qm.ini” a pagina 109](#)

Un riepilogo degli attributi delle stanze del file di configurazione del gestore code, `qmi.ini`, con link a ulteriori informazioni.

Riepilogo delle stanze e degli attributi del file qm.ini

Un riepilogo degli attributi delle stanze del file di configurazione del gestore code, `qmi.ini`, con link a ulteriori informazioni.

Tabella 12. Stanze del file qm.ini

Stanza e attributi	Descrizione degli attributi
Windows StanzaAccessMode	
Windows <u>gruppo di accesso</u> ¹	Un gruppo di protezione Windows , i cui membri avranno accesso completo a tutti i file di dati del gestore code.
ApiExitStanza locale	
<u>Nome</u>	Il nome descrittivo dell'uscita API inoltrato nel campo Nome ExitInfodella struttura MQAXP.
<u>funzione</u>	Il nome del punto di ingresso della funzione nel modulo contenente il codice di uscita API.
<u>Modulo</u>	Il modulo contenente il codice di uscita API.
<u>Dati</u>	I dati da passare all'uscita API nel campo ExitData della struttura MQAXP.
<u>Sequenza</u>	La sequenza in cui questa uscita API viene richiamata rispetto ad altre uscite API.
V 9.4.0 Linux AIX Stanza AuthToken	
<u>KeyStore</u>	Percorso file per il keystore che contiene i certificati di chiave pubblica o le chiavi simmetriche dell'emittente attendibile.
<u>KeyStorePwdFile</u>	Percorso file per il file che contiene la password codificata per il keystore.
<u>CertLabel</u>	L'etichetta del certificato per un certificato di chiave pubblica o simmetrica nel keystore utilizzato per convalidare i token di autenticazione.
<u>UserClaim</u>	Richiedere all'interno del token che contiene le informazioni sull'identità dell'utente che il gestore code può adottare per i controlli di autorizzazione.
<u>AllowOSGroups</u>	Questo attributo determina se l'appartenenza al gruppo per l'utente adottato è selezionata o meno.
Stanza AutoCluster	
<u>Tipo</u>	Il tipo di cluster automatico. L'unica opzione valida è Uniform, che rappresenta un cluster uniforme.
<u>ClusterName</u>	Il nome del cluster automatico.
<u>RepositoryName1</u>	Il nome del gestore code del primo repository completo nel cluster automatico.
<u>Repository1Conname</u>	Il valore del nome connessione (CONNNAME) per il modo in cui i membri del cluster automatico devono connettersi al gestore code.
<u>RepositoryName2</u>	Il nome del gestore code per il secondo repository completo nel cluster automatico.
<u>Repository2Conname</u>	Il valore del nome connessione (CONNNAME) per il modo in cui i membri del cluster automatico devono connettersi al gestore code.

Tabella 12. Stanze del file qm.ini (Continua)

Stanza e attributi	Descrizione degli attributi
Stanza AutoConfig	
<u>Configurazione MQSC</u>	Un percorso file completo o un percorso a una directory, in cui tutti i file *.mqsc vengono applicati al gestore code ad ogni avvio del gestore code.
<u>IniConfig</u>	Un percorso file completo o un percorso a una directory, in cui tutti i file *.ini vengono applicati al file qm.ini ad ogni avvio del gestore code.
Stanza Canali	
<u>MaxChannels</u>	Il numero massimo di canali correnti consentiti.
<u>MaxActiveChannels</u>	Il numero massimo di canali che possono essere attivi in qualsiasi momento.
<u>MaxInitiators</u>	Il numero massimo di iniziatori.
<u>MQIBindType</u>	Il binding per le applicazioni.
<u>PipeLineLength</u>	Il numero massimo di thread simultanei che un canale utilizzerà.
<u>AdoptNewMCA</u>	Quali tipi di canali possono avere l'istanza del canale esistente arrestata in modo che una nuova istanza del canale possa essere avviata quando IBM MQ riceve una richiesta di avviare un canale, ma rileva che un'istanza del canale è già in esecuzione.
<u>AdoptNewMCATimeout</u>	La quantità di tempo, in secondi, per cui la nuova istanza del canale attende la fine della vecchia.
<u>AdoptNewMCACheck</u>	Il tipo di controllo richiesto quando si abilita l'attributo AdoptNewMCA .
<u>ChlauthEarlyAdottare</u>	L'ordine in cui vengono elaborate le regole di autenticazione della connessione e di autenticazione del canale.
<u>PasswordProtection</u>	Indica se le credenziali specificate da un'applicazione devono essere protette con la protezione della password MQCSP, se il canale non utilizza la codifica TLS.
<u>IgnoreSeqNumberMismatch</u>	Controlla come il gestore code gestisce una mancata corrispondenza di numeri di sequenza durante l'avvio del canale.
Stanza Connessione	
<u>DefaultBindDefaultBind</u>	Se le applicazioni e il gestore code, che vengono eseguiti in processi separati, condividono alcune risorse o nessuna risorsa tra loro.
Stanza DiagnosticMessages	
<u>Nome</u>	Nome di una stanza.
<u>Servizio</u>	Un servizio che viene abilitato da questa sezione.
<u>ExcludeMessage</u>	I messaggi che non devono essere scritti nel log degli errori del gestore code.

Tabella 12. Stanze del file `qm.ini` (Continua)












Stanza e attributi	Descrizione degli attributi
SuppressMessage	Messaggi che devono essere scritti nel log degli errori del gestore code una sola volta in un intervallo di tempo specificato.
 SuppressInterval	L'intervallo di tempo, in secondi, in cui i messaggi specificati in SuppressMessage vengono scritti una sola volta nel log degli errori del gestore code.
Gravità	Un elenco separato da virgole di livelli di severità.
FilePath	Il percorso in cui vengono scritti i file di log. (Supportato solo quando l'attributo Servizio è impostato su File.)
FilePrefix	Il prefisso dei file di log. (Supportato solo quando l'attributo Servizio è impostato su File.)
FileSize	La dimensione alla quale il log esegue il rollover. (Supportato solo quando l'attributo Servizio è impostato su File.)
formato	Il formato del file. (Supportato solo quando l'attributo Servizio è impostato su File.)
  Syslog	Il servizio Syslog che invia tutti i messaggi non filtrati a syslog utilizzando la specifica dei messaggi di diagnostica formato JSON .
  Rientro	Il valore ident associato alle voci syslog. Supportato solo quando l'attributo Servizio è impostato su Syslog.
Stanza ExitPath	
ExitsDefaultPath	Il percorso per i programmi di uscita utente sul sistema del gestore code (32 bit).
ExitsDefaultPath64	Il percorso per i programmi di uscita utente sul sistema del gestore code (64 bit).
ExitPropertiesstanza locale	
CLWLMode	Indica se l'uscita CLWL (cluster workloac) viene eseguita in modalità FAST o SAFE.
   Stanza del file system	
   ValidateAuth	Consentire agli utenti che non sono membri del gruppo mqm di accedere a file e directory di errori.
Stanza di log	
LogPrimaryFiles	I file di log assegnati quando viene creato il gestore code.
LogSecondaryFiles	I file di log assegnati quando i file primari sono esauriti.
LogFilePages	Il numero di pagine del file di log. (La dimensione del file di log è specificata in unità di pagine da 4 KB.)
LogType	Il tipo di registrazione che deve essere utilizzato dal gestore code (circolare o lineare).

Tabella 12. Stanze del file qm.ini (Continua)

Stanza e attributi	Descrizione degli attributi
LogBufferPages	La quantità di memoria assegnata ai record buffer per la scrittura, specificando la dimensione dei buffer in unità di pagine da 4 KB.
LogPath	La directory in cui risiedono i file di log per un gestore code.
LogWriteIntegrity	Il metodo utilizzato dal programma di registrazione per scrivere in modo affidabile i record di log.
LogManagement	Il metodo utilizzato per gestire le estensioni di log, manualmente o dal gestore code.
Windows StanzaLU62	
Windows Nome TP	Il nome TP da avviare sul sito remoto.
Windows Library1	Il nome della DLL APPC.
Windows Library2	Lo stesso di Library1, utilizzato se il codice è memorizzato in due librerie separate.
CP4I StanzaNativeHAInstance	
"Nome" a pagina 146	Il nome istanza utilizzato quando è stata creata l'istanza del gestore code.
"ReplicationAddress" a pagina 146	Il nome host, il numero decimale puntato IPv4 o l'indirizzo in formato esadecimale IPv6 dell'istanza.
CP4I NativeHALocalStanza istanza	
"LocalName" a pagina 146	Il nome della stanza dell'istanza NativeHALocal, preso dal nome dell'istanza della replica del log specificato quando viene creato il gestore code della HA nativa.
"KeyRepository" a pagina 146	L'ubicazione del repository delle chiavi che contiene il certificato digitale da utilizzare per la protezione del traffico di replica del log.
"CertificateLabel" a pagina 147	L'etichetta del certificato che indica il certificato digitale da utilizzare per la protezione del traffico di replica del log.
"CipherSpec" a pagina 147	La CipherSpec di MQ da utilizzare per proteggere il traffico di replica del log.
"LocalAddress" a pagina 147	L'indirizzo dell'interfaccia di rete locale che accetta il traffico di replica del log.
"HeartbeatInterval" a pagina 147	L'intervallo di heartbeat definisce la frequenza in millisecondi con cui un'istanza attiva di un gestore code HA nativo invia un heartbeat di rete.
"HeartbeatTimeout" a pagina 147	Il timeout heartbeat definisce il tempo di attesa di un'istanza di replica di un gestore code HA nativo prima di decidere che l'istanza attiva non risponde.

Tabella 12. Stanze del file qm.ini (Continua)

Stanza e attributi	Descrizione degli attributi
<u>“RetryInterval” a pagina 147</u>	L'intervallo di nuovi tentativi definisce la frequenza in millisecondi con cui un gestore code HA nativo deve ritentare un link di replica non riuscito.
Windows Stanza NETBIOS	
Windows <u>LocalName</u>	Il nome con cui questa macchina è nota sulla LAN.
Windows <u>AdapterNum</u>	Il numero dell'adattatore LAN.
Windows <u>NumSess</u>	Il numero di sessioni da assegnare.
Windows <u>NumCmds</u>	Il numero di comandi da assegnare.
Windows <u>NumNames</u>	Il numero di nomi da assegnare.
Windows <u>Library1</u>	Il nome della DLL NetBIOS.
Stanza QMErrorLog	
<u>DimensioneErrorLog</u>	Specifica la dimensione del log degli errori del gestore code copiato nel backup.
<u>ExcludeMessage</u>	Specifica i messaggi che non devono essere scritti nel log degli errori del gestore code.
<u>SuppressMessage</u>	Specifica i messaggi scritti nel log degli errori del gestore code una sola volta in un intervallo di tempo specificato.
<u>SuppressInterval</u>	Specifica l'intervallo di tempo, in secondi, in cui i messaggi specificati in SuppressMessage vengono scritti nel log degli errori del gestore code una sola volta.
Linux AIX Stanza Modalità limitata ²	
Linux AIX <u>ApplicationGroup</u>	Il nome della coda di trasmissione locale in cui vengono inseriti i messaggi remoti se una coda di trasmissione non è esplicitamente definita per la relativa destinazione.
Stanza di protezione	
<u>ClusterQueueAccessControl</u>	Controllare il controllo degli accessi delle code cluster o delle code complete ospitate sui gestori code cluster.
Windows <u>GroupModel</u>	Se OAM (Object Authority Manager) controlla i gruppi globali quando determina l'appartenenza al gruppo di un utente su Windows.
Stanza di servizio	
<u>Nome</u>	Il nome del servizio richiesto.
<u>EntryPoints</u>	Il numero di punti di entrata definiti per il servizio.
Windows <u>SecurityPolicy</u>	Su Windows, la politica di sicurezza per ciascun gestore code

Tabella 12. Stanze del file qm.ini (Continua)

Stanza e attributi	Descrizione degli attributi
<div style="display: flex; align-items: center;"> <div style="background-color: #4F7942; color: white; padding: 2px 5px; margin-right: 5px;">Linux</div> <div style="background-color: #4F7942; color: white; padding: 2px 5px; margin-right: 5px;">AIX</div> <div style="border-bottom: 1px solid black; padding-left: 5px;">SecurityPolicy</div> </div>	<p>Su AIX and Linux, se il gestore code utilizza l'autorizzazione basata sull'utente o sul gruppo.</p> <p>Da IBM MQ 9.3.0, è inoltre possibile creare un nome utente non operativo.</p>
<u>SharedBindingsUserId</u>	Solo per i binding condivisi, se il campo UserIdentifier nella struttura IdentityContext , dalla funzione MQZ_AUTHENTICATE_USER, è l'ID utente effettivo o l'ID utente reale.
<u>FastpathBindingsUserId</u>	Solo per i collegamenti fastpath, se il campo UserIdentifier nella struttura IdentityContext , dalla funzione MQZ_AUTHENTICATE_USER, è l'ID utente effettivo o l'ID utente reale.
<u>IsolatedBindingsUserId</u>	Solo per bind isolati, se il campo UserIdentifier nella struttura IdentityContext , dalla funzione MQZ_AUTHENTICATE_USER, è l'ID utente effettivo o l'ID utente effettivo.
Stanza ServiceComponent	
<u>Servizio</u>	Il nome del servizio richiesto.
<u>Nome</u>	Il nome descrittivo del componente servizio.
<u>Modulo</u>	Il nome del modulo che deve contenere il codice per questo componente.
<u>ComponentDataComponentData</u>	La dimensione, in byte, dell'area dati del componente passata al componente su ciascuna chiamata.
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> Stanza SPX	
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> <u>Socket</u>	Il numero di socket SPX in notazione esadecimale.
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> <u>BoardNum</u>	Il numero dell'adattatore LAN.
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> <u>KeepAlive</u>	Attivare o disattivare la funzione KeepAlive .
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> <u>Library1</u>	Il nome della DLL SPX.
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> <u>Library2</u>	Lo stesso di LibraryName1, utilizzato se il codice è memorizzato in due librerie separate.
<div style="background-color: #4F7942; color: white; padding: 2px 5px; display: inline-block;">Windows</div> <u>ListenerBacklog</u>	Sovrascrivere il numero predefinito di richieste in sospeso per il listener SPX.
Stanza SSL	
<u>OutboundSNI</u>	Specifica se i client con capacità SNI imposteranno SNI sul nome del canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS o sul nome host.

Tabella 12. Stanze del file qm.ini (Continua)







Stanza e attributi	Descrizione degli attributi
<u>AllowOutbound</u> <u>AllowOutbound</u>	<p>Specifica se i client con capacità SNI imposteranno SNI sul nome canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS.</p> <p> Attenzione:  Da IBM MQ 9.3.0 questa proprietà è obsoleta. Utilizzare invece OutboundSNI.</p>
<u>AllowedCipher</u>	<p>Specifica un elenco personalizzato di CipherSpecs ordinati e abilitati per l'utilizzo con canali IBM MQ su Multiplatforms.</p>
<u>AllowTLSV13</u>	<p>Se un gestore code è in grado di utilizzare TLS 1.3 CipherSpecs.</p>
<u>CDPCheckExtensions</u>	<p>Se i canali TLS su questo gestore code tentano di controllare i server CDP denominati nelle estensioni del certificato del punto CrlDistribution.</p>
<u>MinimumRSAKeyDimensione</u>	<p>Specifica la dimensione chiave minima che i certificati RSA devono avere per essere accettati.</p>
<u>OCSPAAuthentication</u>	<p>L'azione da intraprendere quando uno stato di revoca non può essere determinato da un server OCSP.</p>
<u>OCSPCheckExtensions</u>	<p>Indica se i canali TLS su questo gestore code tentano di controllare i server OCSP denominati nelle estensioni certificato di accesso AuthorityInfo.</p>
<u>OCSPTimeout</u>	<p>Il numero di secondi di attesa per un responder OCSP durante l'esecuzione di una verifica di revoca.</p>
 <u>PeerCertChainValidation</u>	<p>L'impostazione di convalida del certificato IBM Global Security Kit (GSKit).</p>
<u>SSLHTTPProxyName</u>	<p>Il nome host o l'indirizzo di rete del server proxy HTTP che deve essere utilizzato da GSKit per i controlli OCSP.</p>
<u>SSLHTTPConnectTimeout</u>	<p>Il numero di secondi di attesa per stabilire correttamente una connessione di rete a un server HTTP durante l'esecuzione di un controllo di revoca.</p>
Stanza del pool secondario "3" a pagina 118	<p>Questa stanza è creata da IBM MQ. Non modificarla.</p>
<u>ShortSubpool</u> <u>ShortSubpool</u> "3" a pagina 118	<p>Un nome corrispondente a una directory e a un collegamento simbolico creato all'interno della directory /var/mqm/sockets, che IBM MQ utilizza per le comunicazioni interne tra i propri processi in esecuzione.</p>
 Stanza TCP	
<u>PORT</u>	<p>Il numero di porta predefinito, in notazione decimale, per le sessioni TCP/IP.</p>
 <u>Library1</u>	<p>Il nome della DLL socket TCP/IP.</p>
<u>KeepAlive</u>	<p>Attivare o disattivare la funzione KeepAlive.</p>
<u>ListenerBacklog</u>	<p>Sovrascrivere il numero predefinito di richieste in sospeso per il listener TCP/IP.</p>

Tabella 12. Stanze del file *qm.ini* (Continua)

Stanza e attributi	Descrizione degli attributi
Timeout connessione	Il numero di secondi prima del timeout di un tentativo di connessione del socket.
SndBuffSndBuff	La dimensione in byte del buffer di invio TCP/IP utilizzato dalla fine di invio dei canali.
RcvBuff	La dimensione in byte del buffer di ricezione TCP/IP utilizzato dall'estremità di ricezione dei canali.
RcvSndBuffSize	La dimensione in byte del buffer di invio TCP/IP utilizzato dall'estremità mittente di un canale ricevente.
RcvRcvBuffSize	La dimensione in byte del buffer di ricezione TCP/IP utilizzato dall'estremità di ricezione di un canale ricevente.
SvrSndBuffSize	La dimensione in byte del buffer di invio TCP/IP utilizzato dall'estremità del server di un canale di connessione server di connessione client.
SvrRcvBuffSize	La dimensione in byte del buffer di ricezione TCP/IP utilizzato dall'estremità server di un canale di connessione server di connessione client.
 SecureCommsSolo	Specifica se è consentita la comunicazione in testo semplice, il valore predefinito o non consentito.
Stanza dei parametri di ottimizzazione	
SuppressDspAuthFail	Se il gestore code elimina la generazione di eventi di autorizzazione e la scrittura di messaggi di errore AMQ8077 nel log degli errori quando un controllo di autorizzazione non riesce, se la connessione non dispone dell'autorizzazione + dsp per un oggetto.
ImplSyncOpenOutput	Il numero minimo di applicazioni che hanno la coda aperta per l'inserimento, prima che un punto di sincronizzazione implicito possa essere abilitato per un'operazione di inserimento permanente, al di fuori del punto di sincronizzazione.
NomeUniformCluster	Il nome del cluster di IBM MQ che si sta utilizzando come cluster uniforme.
OAMLdapConnectTimeout	Il tempo massimo, in secondi, che il client LDAP attenderà per stabilire una connessione TCP al server.
OAMLdapQueryTimeLimit	Il tempo massimo, in secondi, che il client LDAP attende per ricevere una risposta a una richiesta LDAP dal server.
OAMLdapResponseWarningTime	Se una connessione a un server LDAP ha impiegato più tempo del numero di soglia di secondi specificato dal parametro OAMLdapResponseWarningTime , un messaggio AMQ5544W verrà scritto nel log degli errori.
ExpiryInterval	Indica la frequenza con cui il gestore code esegue la scansione delle code alla ricerca di messaggi scaduti che non sono stati già ripuliti da altre attività della coda. Si tratta di un intervallo di tempo in secondi.

<i>Tabella 12. Stanze del file qm.ini (Continua)</i>	
Stanza e attributi	Descrizione degli attributi
<u>LivenessHeartBeatLen</u>	Configura la frequenza con cui i controlli del gestore code che scrivono nel log vengono eseguiti ad una velocità ragionevole.
<u>ECHeartBeatLen</u>	Configura la frequenza dei controlli di integrità generali del gestore code.
<u>FileLockHeartBeatLen</u>	Modifica il valore predefinito per i controlli del blocco file per un gestore code a più istanze che il controller di esecuzione esegue periodicamente per assicurarsi che mantenga ancora il blocco esclusivo sul file a più istanze primario.
Stanza Variabili	
<i>attributo=valore</i>	Un nome e un valore associato da utilizzare come inserimento durante le definizioni MQSC.
Stanza XAResourceManager	
<u>Nome</u>	L'istanza del gestore risorse.
<u>SwitchFile</u>	Il nome completo del file di caricamento contenente la struttura dello switch XA del gestore risorse.
<u>XAOpenString</u>	La stringa di dati da passare al punto di ingresso xa_open del gestore risorse.
<u>XACloseString</u>	La stringa di dati da trasmettere al punto di ingresso xa_close del gestore risorse.
<u>ThreadOfControl</u>	Il valore che il gestore code utilizza per la serializzazione quando deve richiamare il gestore risorse da uno dei propri processi a più thread. Obbligatorio per Windows.

Note:

1. La stanza AccessMode viene impostata dall'opzione **-a [r]** nel comando **crtmqm**. Non modificare la stanza AccessMode dopo che il gestore code è stato creato.
2. La stanza RestrictedMode è impostata dall'opzione **-g** nel comando **crtmqm**. Non modificare questa stanza dopo la creazione del gestore code. Se non si utilizza l'opzione **-g**, la stanza non viene creata nel file qm.ini.
3. La stanza Subpool e il nome dell'attributo ShortSubpool all'interno di tale stanza, vengono scritti automaticamente da IBM MQ quando si crea un gestore code. IBM MQ sceglie un valore per il nome ShortSubpool. Non modificare questo valore.

Windows Stanza AccessMode del file qm.ini

La modalità di accesso si applica solo a server Windows. La stanza AccessMode del file qm.ini è impostata dall'opzione **-a [r]** sul comando **crtmqm**. Non modificare la stanza AccessMode dopo che il gestore code è stato creato.

Utilizzare il gruppo di accessi (**-a [r]**) opzione del comando **crtmqm** per specificare un gruppo di sicurezza Windows, ai cui membri verrà concesso l'accesso completo a tutti i file di dati del gestore code. Il gruppo può essere un gruppo locale o globale, a seconda della sintassi utilizzata. La sintassi valida per il nome gruppo è la seguente:

LocalGroup
Nome dominio\GlobalGroup nome

Nome GlobalGroup @ Nome dominio

È necessario definire il gruppo di accesso aggiuntivo prima di eseguire il comando **crtmqm** con l'opzione **-a [r]**.

Se si specifica il gruppo utilizzando **-ar** invece di **-a**, al gruppo mqm locale non viene concesso l'accesso ai file di dati del gestore code. Utilizzare questa opzione se il file system che ospita i file di dati del gestore code non supporta le voci di controllo accessi per i gruppi definiti localmente.

Il gruppo è in genere un gruppo di sicurezza globale, utilizzato per fornire ai gestori code a più istanze l'accesso a una cartella condivisa con log e dati dei gestori code. Utilizzare il gruppo di accesso di sicurezza aggiuntivo per impostare le autorizzazioni di lettura e scrittura sulla cartella o per condividere i file di log e i dati del gestore code in essa contenuti.

Il gruppo di accesso di sicurezza aggiuntivo è un'alternativa all'utilizzo del gruppo locale denominato mqm per impostare le autorizzazioni sulla cartella contenente i log e i dati del gestore code. A differenza del gruppo locale mqm, è possibile impostare il gruppo di accesso di sicurezza aggiuntivo come un gruppo locale o globale. Deve essere di tipo globale per impostare le autorizzazioni sulle cartelle condivise che contengono i dati e i file di log utilizzati dai gestori code a più istanze.

Il sistema operativo Windows controlla le autorizzazioni di accesso necessarie per leggere e scrivere i dati e i file di log del gestore code. Controlla le autorizzazioni dell'ID utente che esegue i processi dei gestori code. L'ID utente controllato dipende a seconda che il gestore code sia stato avviato come servizio o in modo interattivo. Se si è avviato il gestore code as-a-service, l'ID utente controllato dal sistema Windows è l'ID utente configurato con la procedura guidata **Prepara IBM MQ**. Se si è avviato il gestore code in modo interattivo, l'ID utente controllato dal sistema Windows è l'ID utente che ha eseguito il comando **strmqm**.

L'ID utente deve essere membro del gruppo mqm locale per avviare il gestore code. Se l'ID utente è membro del gruppo di accesso di sicurezza aggiuntivo, il gestore code può leggere e scrivere i file per cui vengono fornite le autorizzazioni utilizzando il gruppo.

Limitazione: È possibile specificare un gruppo di accesso di sicurezza aggiuntivo solo sui sistemi operativi Windows. Se si specifica un gruppo di accesso di sicurezza aggiuntivo su altri sistemi operativi, il comando **crtmqm** restituisce un errore.

Stanza di esempio

```
AccessMode:  
SecurityGroup=wmq\wmq
```

Concetti correlati

[“Proteggere i file e le directory di log e i dati del gestore code non condivisi su Windows” a pagina 557](#)

[“Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows” a pagina 554](#)

Attività correlate

[“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows” a pagina 529](#)

Riferimenti correlati

[crtmqm \(crea gestore code\)](#)

Multi **ApiExitStanza locale del file qm.ini**

La stanza ApiExitlocale specifica le routine di uscita API per un gestore code.

Per un server, modificare la sezione ApiExitlocale del file `qm.ini` per identificare le routine di uscita API per un gestore code.

Linux **Windows** In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code IBM MQ Explorer Exits.

Per un client, modificare la sezione ApiExitlocale nel file `mqclient.ini` per identificare le routine di uscita API per un gestore code.

Panoramica

La stanza `ApiExitLocal` consente di specificare solo un singolo `Module`, ma è necessario fornire quattro moduli, come segue:

- Senza thread a 32 bit
- A 32 bit con thread
- 64 bit senza thread
- 64 bit con thread

Si noti che IBM MQ accoda `_r` al nome modulo fornito per identificare la versione con thread dell'uscita, ma IBM MQ non fornisce un meccanismo direttamente equivalente per le varianti a 32 bit e 64 bit.

Le versioni di `amqsaxe0` e `amqsaxe0_r` fornite in `prefix/mqm/samp/bin` sono create per la dimensione nativa del gestore code sulla piattaforma per cui sono state create (ora tutte a 64 bit) e possono essere utilizzate solo dalle applicazioni in esecuzione nella stessa dimensione nativa.

Se viene fornito un nome modulo non qualificato, IBM MQ cerca in `/var/mqm/exits` le varianti a 32 bit e in `/var/mqm/exits64` le varianti a 64 bit

Ad esempio, `module=amqsaxe` implica:

```
/var/mqm/exits/amqsaxe - 32 bit unthreaded variant
/var/mqm/exits/amqsaxe_r - 32 bit threaded variant
/var/mqm/exits64/amqsaxe - 64 bit unthreaded variant
/var/mqm/exits64/amqsaxe_r - 64 bit threaded variant
```

Windows Sui sistemi Windows, è anche possibile utilizzare il comando **amqmdain** per modificare le voci per le uscite API. (Per identificare le routine di uscita API per tutti i gestori code, utilizzare le stanze `ApiExitCommon` e `ApiExitTemplate`, come descritto in [“Stanze del modello ApiExitCommon e ApiExitdel file mq.ini”](#) a pagina 101.)

Si noti che, affinché l'uscita API funzioni correttamente, il messaggio dal server deve essere inviato al client non convertito. Dopo che l'uscita API ha elaborato il messaggio, il messaggio deve essere convertito sul client. Ciò, quindi, richiede che siano state installate tutte le uscite di conversione sul client.

Per ulteriori informazioni sull'utilizzo di questi attributi, consultare [Configurazione delle uscite API](#).

Parametri

Nome=ApiExit_name

Il nome descrittivo dell'uscita API inoltrato nel campo `Nome ExitInfo` della struttura `MQAXP`.

Questo nome deve essere univoco, non deve superare i 48 caratteri e contenere solo caratteri validi per i nomi degli oggetti IBM MQ (ad esempio, nomi coda).

Funzione=nome_funzione

Il nome del punto di ingresso della funzione nel modulo contenente il codice di uscita API. Questo punto di ingresso è la funzione `MQ_INIT_EXIT`.

La lunghezza di questo campo è limitata a `MQ_EXIT_NAME_LENGTH`.

Modulo=nome_modulo

Il modulo contenente il codice di uscita API.

Se questo campo contiene il percorso completo del modulo, questo verrà visualizzato così come è. Se questo campo contiene solo il nome del modulo, il modulo si trova utilizzando l'attributo **ExitsDefaultPath** nella stanza `ExitPath` del file `qm.ini`.

Su piattaforme che supportano librerie separate, è necessario fornire sia una versione non con thread che una versione con thread del modulo di uscita API. La versione con thread deve avere un suffisso `_r`. La versione con thread dello stub dell'applicazione IBM MQ accoda implicitamente `_r` al nome modulo fornito prima che venga caricato.

La lunghezza di questo campo è limitata alla lunghezza massima del percorso supportata dalla piattaforma.

Dati=nome_dati

I dati da passare all'uscita API nel campo ExitData della struttura MQAXP.

Se si include questo attributo, vengono rimossi gli spazi iniziali e finali, la stringa rimanente viene troncata a 32 caratteri e il risultato viene passato all'uscita. Se si omette questo attributo, il valore predefinito di 32 spazi viene passato all'uscita.

La lunghezza massima di questo campo è 32 caratteri.

Sequenza=numero_sequenza

La sequenza in cui questa uscita API viene richiamata rispetto ad altre uscite API. Un'uscita con un numero di sequenza basso viene richiamata prima di un'uscita con un numero di sequenza più alto. Non è necessario che la numerazione di sequenza delle uscite sia contigua. Una sequenza di 1, 2, 3 ha lo stesso risultato di una sequenza di 7, 42, 1096. Se due uscite hanno lo stesso numero di sequenza, il gestore code decide quale richiamare per primo. È possibile determinare quale elemento è stato richiamato dopo l'evento inserendo l'ora o un indicatore nell'area ExitChainindicata da ExitChainAreaPtr in MQAXP oppure scrivendo il proprio file di log.

Questo attributo è un valore numerico senza segno.

Stanza di esempio

```
ApiExitLocal:  
  Name=ClientApplicationAPIChecker  
  Sequence=3  
  Function=EntryPoint  
  Module=/usr/Dev/ClientAppChecker  
  Data=9.20.176.20
```

V 9.4.0

Linux

AIX

Stanza AuthToken del file qm.ini

Utilizzare la stanza **AuthToken** per configurare il gestore code per convalidare i token di autenticazione forniti dalle applicazioni di connessione. Se il servizio di autenticazione supporta un endpoint JWKS per la configurazione della chiave, questa è di solito un'opzione preferibile.

Per ulteriori informazioni, consultare [Configurazione di un gestore code per accettare i token di autenticazione utilizzando un endpoint JWKS](#).

Stanza AuthToken

KeyStore= stringa

Percorso file per il keystore che contiene i certificati della chiave pubblica dell'emittente attendibile e le chiavi simmetriche. È possibile aggiungere le chiavi a un keystore esistente o creare un nuovo keystore. Per ulteriori informazioni, consultare [Configurazione di un gestore code per accettare i token di autenticazione](#). Il gestore code utilizza le chiavi nel keystore per verificare che il token di autenticazione presentato dall'applicazione sia firmato dall'emittente attendibile.

È possibile utilizzare un keystore CMS con estensione file `.kdb` oppure un keystore PKCS#12 con estensione file `.p12`. Se il file keystore non esiste o non è possibile accedervi, un errore AMQ7076E: Valore non valido per l'attributo nel file ini viene emesso nel log degli errori del gestore code.

Verificare che il tipo di keystore corrisponda all'estensione del nome file per il keystore. IBM MQ rileva il formato corretto del keystore, le incongruenze potrebbero causare altri problemi di gestione, tuttavia, se il tipo di keystore e l'estensione del nome file non corrispondono.

La lunghezza massima del percorso del file keystore è 256 caratteri.

KeyStorePwdFile= stringa

Percorso file per il file che contiene la password codificata per il keystore. Il file deve contenere la password codificata come una singola riga di testo. Le password in testo semplice non sono accettate.

Utilizzare il comando **runqmcrcd** per codificare la password prima di salvarla nel file di password del keystore. Il file di password del keystore deve contenere solo la password codificata creata eseguendo il comando **runqmcrcd**.

La lunghezza massima della password in testo semplice prima che venga codificata è di 1024 caratteri.

Questo parametro è facoltativo. Se non viene fornito, il gestore code ricerca un file stash con la password nella stessa directory e con lo stesso nome del keystore, ma con l'estensione file `.sth`. Se il file di stash non viene trovato, la configurazione viene rifiutata e il messaggio di errore AMQ7006E viene emesso nel log degli errori del gestore code. Per ulteriori informazioni sulle opzioni per memorizzare le password del keystore, consultare [Codifica delle password del repository delle chiavi](#).

La lunghezza massima del percorso del file di password è 256 caratteri.

CertLabel= stringa

L'etichetta del certificato per un certificato di chiave pubblica o simmetrica nel keystore utilizzato per convalidare i token di autenticazione. È possibile fornire fino a 32 etichette di certificato ripetendo l'attributo **CertLabel**.

Quando si aggiungono i certificati al keystore del gestore code, fornire loro etichette significative. Le etichette del certificato sono sensibili al maiuscolo / minuscolo. Possono contenere caratteri alfanumerici, caratteri di punteggiatura e spazi. Se viene rilevato un carattere non valido, viene restituito un errore e viene scritto un messaggio di errore nel log degli errori IBM MQ.

Gli emittenti di token attendibili potrebbero fornire più certificati di chiave pubblica e chiavi simmetriche. Ad esempio, i certificati di chiave pubblica hanno periodi di validità. Quando sono prossimi alla scadenza, l'emittente del token fornisce un nuovo certificato con una nuova data di scadenza. Per un certo periodo, entrambi i certificati potrebbero essere validi.

Quando le applicazioni presentano token per l'autenticazione, l'elenco di **CertLabels** viene controllato fino a quando non viene trovata una chiave valida utilizzata per firmare il token. Se la corrispondenza viene trovata, la firma del token viene convalidata.

Se **CertLabel** non viene specificato, la connessione dall'applicazione che presenta il token ha esito negativo con il codice di errore 2063 MQRC_SECURITY_ERROR e il messaggio AMQ5786E: Errore di configurazione del token di autenticazione viene scritto nel log degli errori del gestore code.

La lunghezza massima dell'etichetta del certificato è 64 caratteri.

Ad esempio:

```
AuthToken:
  KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb
  KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
  CertLabel=token
  CertLabel=rsakey
  CertLabel=mark
  ... up to 32 CertLabel fields
```

UserClaim= stringa

Richiesta all'interno del token che contiene l'ID utente che il gestore code adotta per i controlli di autorizzazione.

Questo parametro è facoltativo se il gestore code è configurato con **ADOPTCTX(NO)**. Se si utilizza **ADOPTCTX(YES)**, questo parametro è obbligatorio. **ADOPTCTX** è un attributo presente nell'oggetto delle informazioni di autenticazione (AUTHINFO) a cui fa riferimento l'attributo **CONNAUTH** del gestore code.

Per adottare un'identità, il token deve contenere un'asserzione con il nome specificato nell'attributo **UserClaim** della stanza **AuthToken** e deve essere utilizzato **ADOPTCTX(YES)**.

Ad esempio, se il tuo token include una richiesta "AppUser": "MyUserName", devi specificare `UserClaim=AppUser` nella sezione `AuthToken` del file `qm.ini` per adottare l'identità "MyUserName" per l'autorizzazione.

La lunghezza massima del valore dell'attributo **UserClaim** è 128 caratteri.

Nota: Da IBM MQ 9.4.0, se viene specificata la stanza AuthToken , il valore effettivo dell'attributo **SecurityPolicy** della stanza Service è impostato su UserExternal. L'autenticazione del token non è disponibile se **SecurityPolicy** è esplicitamente impostato su Gruppo nella stanza Service. Se **SecurityPolicy** è impostato su Gruppo, rimuovere l'attributo **SecurityPolicy** dalla sezione Servizio, quindi riavviare il gestore code. Per ulteriori informazioni, consultare [SecurityPolicy](#).

Nota: Utilizzare l'attributo **ADOPTCTX** dell'oggetto delle informazioni di autenticazione per controllare se l'ID utente nel token viene adottato per i controlli di autorizzazione. Quando si crea il gestore code, questo attributo è impostato su **ADOPTCTX (YES)**. Questo valore fa sì che venga adottato l'ID utente dal token. L'ID utente deve rispettare i requisiti per gli ID utente nei token di autenticazione. Per ulteriori informazioni, vedi [ID utente nei token di autenticazione](#). Se la richiesta utente token contiene un ID utente che non soddisfa i requisiti, la connessione viene rifiutata con il codice di errore **2035 MQRC_NOT_AUTHORIZED**. Se **ADOPTCTX (NO)** è impostato, il token viene utilizzato solo per l'autenticazione e un altro utente deve essere utilizzato per l'autorizzazione.

AllowOSGroups=NO (predefinito) |YES

Il valore predefinito è No. Determina se un'identità adottata da un token viene considerata come un utente del sistema operativo e se le appartenenze al gruppo dell'utente del sistema operativo corrispondente vengono rispettate durante l'autorizzazione.

AllowOSGroups= NO | N

I controlli di autorizzazioni si basano solo sul nome dell'utente che viene adottato dal token.

AllowOSGroups= Sì | Y

I controlli di autorizzazione si basano sul nome dell'utente e vengono controllati anche i gruppi a cui potrebbero appartenere.

Nota: L'attributo **allowOSGroups** della sezione **AuthToken** è ancora attivo durante l'autenticazione del token, anche se tutte le altre configurazioni di convalida del token vengono gestite tramite la stanza **JWKS** .

Stanza di esempio - solo autenticazione

È valido che questa stanza contenga solo l'attributo **AllowOSGroups** . Tuttavia, se la configurazione del keystore locale viene inclusa, deve contenere come minimo:

- Percorso file **KeyStore** e
- Nome **CertLabel** .

```
AuthToken:  
  KeyStore=/var/mqm/qmgrs/qmgrs/qm1/tokenissuer/key.kdb  
  CertLabel=token  
  ... up to 32 CertLabel fields
```

Se sono stati inclusi solo i due parametri minimi:

- Un file stash key . sth deve esistere con la password del keystore codificato in modo che il file di password del keystore non sia richiesto.
- Il token non contiene un nome utente da passare a IBM MQ per l'autorizzazione. L'applicazione può connettersi ed essere autenticata, ma deve essere presente un meccanismo diverso per fornire l'autorizzazione affinché l'applicazione possa eseguire il lavoro dopo che è stata connessa.

A seconda della configurazione del tuo gestore code, il nome utente utilizzato per l'autorizzazione potrebbe essere quello definito sul canale tramite regole MCA o il nome utente che l'applicazione client ha eseguito come potrebbe esistere sul tuo server e appartenere a gruppi con autorizzazioni. Tenere presente che quando si utilizzano i token:

- Il gestore code viene inserito in modalità **UserExternal** , il che significa che gli utenti che non esistono sul sistema operativo su cui è in esecuzione il gestore code possono essere utilizzati per l'autenticazione.
- Anche se non si include l'opzione **AllowOSGroups** nella stanza **AuthToken** qm . ini , il valore predefinito è impostato su No. Pertanto, se si include **UserClaim** ma non si specifica

AllowOSGroups=Yes, l'utente del token adottato per l'autorizzazione non viene controllato per i gruppi a cui potrebbe appartenere sul sistema operativo su cui è in esecuzione il gestore code.

Stanza di esempio - autenticazione e autorizzazione

È possibile definire tutti i parametri **AuthToken** :

- Percorso file **KeyStore** ,
- Percorso file **KeyStorePwdFile** ,
- Nome **CertLabel** ,
- **UserClaim** nome e
- **AllowOSGroups** .

```
AuthToken:  
  KeyStore=/var/mqm/qmgrs/qmgrs/QMJWT/ssl/key.kdb  
  KeyStorePwdFile=/var/mqm/qmgrs/QMJWT/ssl/key.pw  
  CertLabel=token  
  CertLabel=rsakey  
  CertLabel=mark  
  ... up to 32 CertLabel fields  
  UserClaim=AppUser  
  AllowOSGroups=Y
```

Se sono stati inclusi tutti i parametri disponibili:

- Codificare la password per il keystore utilizzando il comando **runqmcrcd** . Salvarlo in un file e includere il percorso file nella stanza **AuthToken** .
- Il nome utente presente nella richiesta utente del token di autenticazione viene utilizzato sia per l'autenticazione che per l'autorizzazione.
 - L'utente token potrebbe esistere come utente sul sistema operativo su cui è in esecuzione il gestore code.
 - È stato definito un oggetto delle informazioni di autenticazione per abilitare il controllo utente.
 - Impostare i record di autenticazione di canale per adottare un utente con autorizzazione per interagire con gli oggetti IBM MQ , in base all'autenticazione di canale o alle regole MCA.

La strategia per l'autenticazione e l'autorizzazione degli utenti token dipende dai requisiti e da come i gestori code IBM MQ sono già configurati. Per ulteriori informazioni, vedi [Gestione dei token di autenticazione](#).

Concetti correlati

[Utilizzo dei token](#)

Attività correlate

[Configurazione di un gestore code per accettare AuthTokens](#)

[Utilizzo dei token di autenticazione in una applicazione](#)

Multi

Stanza AutoCluster del file qm.ini

La stanza AutoCluster viene utilizzata quando il gestore code viene avviato per identificare se il cluster è un membro di un cluster automatico e può identificare i repository completi del cluster.

I seguenti attributi sono obbligatori per la stanza AutoCluster :

Tipo = *Uniforme*

Specifica il tipo di cluster automatico e l'unica opzione valida è *Uniforme*, che rappresenta un cluster uniforme.

ClusterName=< *Stringa*>

Il nome del cluster, ovvero il nome del cluster automatico.

I seguenti attributi sono facoltativi per la stanza AutoCluster , ma è necessario fornirli in coppie:

RepositoryName1 =< Stringa>

Questo è il nome del gestore code per il primo repository completo nel cluster automatico. Può essere il nome di questo gestore code o di un altro gestore code.

Repository1Conname=< Stringa nome connessione>

Questo è il valore del nome connessione (CONNNAME) per il modo in cui i membri del cluster automatico devono connettersi a questo gestore code.

Repository2Name=< Stringa>

Questo è il nome del gestore code per il secondo repository completo nel cluster automatico. Può essere il nome di questo gestore code o di un altro gestore code.

Repository2Conname=< Stringa nome connessione>

Questo è il valore del nome connessione (CONNNAME) per il modo in cui i membri del cluster automatico devono connettersi a questo gestore code.

Stanza di esempio

```
AutoCluster:
  Repository1Name=QM1
  Repository2Name=QM2
  Repository1Conname=127.0.0.1(1414)
  Repository2Conname=127.0.0.1(1415)
  ClusterName=UNIFORMCLUSTER1
  Type=Uniform
```

Concetti correlati

[“Bilanciamento automatico dell'applicazione” a pagina 428](#)

Il bilanciamento automatico delle applicazioni migliora notevolmente la distribuzione e la disponibilità delle applicazioni, consentendo a un cluster uniforme IBM MQ di gestire da vicino la distribuzione delle applicazioni nel cluster, piuttosto che affidarsi alla randomizzazione o al blocco manuale delle applicazioni a specifici gestori code.

Attività correlate

[“Creazione di un nuovo cluster uniforme” a pagina 442](#)

Come creare un nuovo cluster uniforme.

Riferimenti correlati

[“Utilizzo della configurazione automatica del cluster” a pagina 446](#)

Configurare IBM MQ per abilitare la configurazione automatica modificando le informazioni di configurazione `qm.ini`.

Multi Stanza AutoConfig del file qm.ini

Gli attributi della stanza AutoConfig vengono utilizzati frequentemente come parte dell'impostazione di cluster uniformi.

Nota: È possibile utilizzare la stanza AutoCluster solo per i cluster uniformi.

MQSCConfig=< Path>

Il percorso è un percorso file completo o un percorso a una directory, in cui tutti i file `*.mqsc` vengono applicati al gestore code ad ogni avvio del gestore code.

Per ulteriori informazioni, consultare [Configurazione automatica da uno script MQSC all'avvio](#).

IniConfig=< Path>

Il percorso è un percorso file completo o un percorso a una directory, in cui tutti i file `*.ini` vengono applicati al file `qm.ini` ad ogni avvio del gestore code.

Per ulteriori informazioni, consultare [“Configurazione automatica di qm.ini all'avvio” a pagina 108](#).

ConfigTimeout

Il valore (in secondi) che il gestore code attende per il completamento della configurazione automatica. Dopo tale periodo di tempo, il gestore code continua l'avvio ed è disponibile per la connessione delle applicazioni.

Il comportamento predefinito è nessun timeout. Ciò significa che il gestore code non sarà disponibile per la connessione delle applicazioni, fino a quando non saranno stati completati tutti i comandi di configurazione automatica.

Non si deve configurare questo attributo semplicemente perché la configurazione sta impiegando molto tempo, perché le applicazioni potrebbero essere in grado di connettersi prima che la configurazione che si applica ad esse sia stata completata, ad esempio, la creazione di code necessarie per l'applicazione.

Stanza di esempio

```
AutoConfig:
MQSCConfig=/tmp/auto.mqsc
IniConfig=/tmp/auto.ini
ConfigTimeout=120
```

Concetti correlati

[“Bilanciamento automatico dell'applicazione”](#) a pagina 428

Il bilanciamento automatico delle applicazioni migliora notevolmente la distribuzione e la disponibilità delle applicazioni, consentendo a un cluster uniforme IBM MQ di gestire da vicino la distribuzione delle applicazioni nel cluster, piuttosto che affidarsi alla randomizzazione o al blocco manuale delle applicazioni a specifici gestori code.

Attività correlate

[“Creazione di un nuovo cluster uniforme”](#) a pagina 442

Come creare un nuovo cluster uniforme.


Riferimenti correlati

[“Utilizzo della configurazione automatica del cluster”](#) a pagina 446



Configurare IBM MQ per abilitare la configurazione automatica modificando le informazioni di configurazione `qm.ini`.

Stanza dei Canali del file `qm.ini`

Gli attributi della stanza Channels determinano la configurazione di un canale.

 Queste informazioni non sono applicabili a IBM MQ for z/OS.

Utilizzare la sezione CHANNELS del file `qm.ini` per specificare informazioni sui canali.

  In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code IBM MQ Explorer Channels.

MaxChannels = 100 (valore predefinito) | numero

Il numero massimo di canali *correnti* consentiti.

Il valore predefinito è 100.

È possibile impostare **MaxChannels** su un valore diverso per limitare il numero massimo di canali correnti, se necessario. Per IBM MQ Appliance, il valore predefinito è 999 999 999 e non deve essere modificato.

MaxActiveChannels = MaxChannels_value

Il numero massimo di canali che possono essere *attivi* in qualsiasi momento. Il valore predefinito è quello specificato per l'attributo **MaxChannels**.

MaxInitiators = 3 (valore predefinito) | numero

Il numero massimo di iniziatori. Il valore predefinito e massimo è 3.

MQBindType= FASTPATH | STANDARD

Il bind per le applicazioni:

Percorso veloce

I canali si collegano utilizzando MQCONNX FASTPATH; nessun processo agent.

STANDARD

I canali si collegano utilizzando STANDARD.

PipeLineLength = 1 | numero

Il numero massimo di thread simultanei che un canale utilizzerà. Il valore predefinito è 1. Qualsiasi valore maggiore di 1 viene considerato come 2.

Quando si utilizza la pipeline, configurare i gestori code ad entrambe le estremità del canale in modo che **PipeLineLength** sia maggiore di 1.

Nota: Il pipelining è efficace solo per i canali TCP/IP.

Per ulteriori informazioni, consultare [Supporto di più thread - pipelining](#).

AdoptNewMCA = NO (valore predefinito) | SVR | SDR | RCVR | CLUSRCVR | ALL | FASTPATH

Se IBM MQ riceve una richiesta di avvio di un canale, ma rileva che un'istanza del canale è già in esecuzione, in alcuni casi l'istanza del canale esistente deve essere arrestata prima che possa essere avviata quella nuova. L'attributo **AdoptNewMCA** permette di controllare quali tipi di canali possono essere terminati in questo modo.

Se si specifica l'attributo **AdoptNewMCA** per un particolare tipo di canale, ma il nuovo canale non si avvia perché un'istanza del canale corrispondente è già in esecuzione:

1. Il nuovo canale tenta di arrestare il canale precedente richiedendone l'arresto.
2. Se il server del canale precedente non risponde a questa richiesta alla scadenza dell'intervallo di attesa **AdoptNewMCATimeout**, il thread o il processo per il precedente server del canale viene terminato.
3. Se il server del canale precedente non è terminato dopo il passo 2 e dopo la scadenza dell'intervallo di attesa **AdoptNewMCATimeout** per una seconda volta, IBM MQ termina il canale con un errore CHANNEL IN USE.

La funzione **AdoptNewMCA** si applica a canali server, mittente, ricevente e ricevente cluster. Nel caso di un canale mittente o server, solo un'istanza di un canale con un determinato nome può essere in esecuzione nel gestore code di ricezione. Nel caso di un canale ricevente o cluster - ricevente, più istanze di un canale con un determinato nome potrebbero essere in esecuzione nel gestore code ricevente, ma solo un'istanza può essere eseguita contemporaneamente da un particolare gestore code remoto.

Nota: AdoptNewMCA non è supportato sui canali di richiesta o di connessione server.

Specificare uno o più valori, separati da virgole o spazi, dal seguente elenco:

No

La funzione **AdoptNewMCA** non è richiesta. Questa è l'opzione predefinita.

SVR

Utilizzare i canali server.

SDR

Utilizzare i canali mittente.

RCVR

Utilizzare i canali riceventi.

CLUSRCVR

Utilizzare i canali riceventi cluster.

TUTTO

Adottare tutti i tipi di canale tranne i canali FASTPATH.

Percorso veloce

Adottare il canale se è un canale FASTPATH. Ciò si verifica solo se viene specificato anche il tipo di canale appropriato, ad esempio: `AdoptNewMCA=RCVR, SVR, FASTPATH`.

Attenzione!: L'attributo MCA `AdoptNew` potrebbe comportarsi in modo imprevedibile con i canali FASTPATH. Prestare particolare attenzione quando si abilita l'attributo MCA `AdoptNew` per i canali FASTPATH.

AdoptNewMCATimeout= 60 (predefinito) | 1-3600

La quantità di tempo, in secondi, per cui la nuova istanza del canale attende la fine della vecchia. Immettere un valore compreso nell'intervallo tra 1 e 3600. Il valore predefinito è 60.

AdoptNewMCACheck = QM | INDIRIZZO | NOME | ALL

Il tipo di controllo richiesto quando si abilita l'attributo `AdoptNewMCA`. Se possibile, eseguire un controllo completo per proteggere i canali dall'arresto, involontario o doloso. Come minimo, verificare che i nomi dei canali corrispondano.

Specificare uno o più dei seguenti valori, separati da virgole o spazi nel caso di *QM*, *NAME* o *ALL*:

QM

Verificare che i nomi dei gestori code corrispondano.

Si noti che il nome del gestore code stesso corrisponde, non il QMID.

ADDRESS

Controllare l'indirizzo IP di origine delle comunicazioni. Ad esempio, l'indirizzo TCP/IP.

Nota: I valori *CONNAME* separati da virgole si applicano agli indirizzi di destinazione e, pertanto, non sono rilevanti per questa opzione.

Nel caso in cui un gestore code a più istanze esegua il failover da *hosta* a *hostb*, i canali in uscita da tale gestore code utilizzeranno l'indirizzo IP di origine *hostb*. Se è diverso da *hosta*, `AdoptNewMCACheck=ADDRESS` non corrisponde.

È possibile utilizzare SSL o TLS con l'autenticazione reciproca per impedire a un aggressore di interrompere un canale in esecuzione esistente. In alternativa, utilizzare una soluzione di tipo HACMP con takeover IP invece di gestori code a più istanze oppure utilizzare un programma di bilanciamento del carico di rete per mascherare l'indirizzo IP di origine.

NOME

Verificare che i nomi dei canali corrispondano.

TUTTO

Controllare i nomi dei gestori code corrispondenti, l'indirizzo di comunicazioni e i nomi dei canali corrispondenti.

Il valore predefinito è `AdoptNewMCACheck=NAME, ADDRESS, QM`.

ChlauthEarlyadotta = Y (predefinito) | N

L'ordine in cui vengono elaborate l'autenticazione della connessione e le regole di autenticazione del canale è un fattore significativo nella determinazione del contesto di sicurezza per connessioni dell'applicazione client IBM MQ.



Attenzione: Il valore predefinito se **ChlauthEarlyAdopt** non è presente nel file `qm.ini` è N. Tuttavia, tutti i gestori code vengono creati con **ChlauthEarlyAdopt=Y** automaticamente aggiunto al file `qm.ini`.

ChlauthEarlyAdopt adotta solo gli ID utente forniti a un gestore code per l'autenticazione della connessione, se *ADOPTCTX* (YES) è impostato sull'oggetto *AUTHINFO* di autenticazione della connessione sul gestore code.

I valori validi per **ChlauthEarlyAdopt** sono i seguenti:

Y

Il canale convalida e adotta le credenziali ID utente e password fornite da un'applicazione che utilizza l'autenticazione della connessione del gestore code prima di applicare le regole di

autenticazione del canale. In questa modalità di funzionamento, le regole di autenticazione del canale corrispondono all'ID utente risultante dai controlli di autenticazione della connessione.

N

Il canale ritarda la convalida di autenticazione della connessione delle credenziali ID utente e password fornite da un'applicazione fino a quando non sono state applicate le regole di autenticazione del canale. Si noti che in questa modalità di operazione, le regole di associazione e di blocco dell'autenticazione del canale non possono considerare i risultati della convalida di ID utente e password.

Ad esempio, l'oggetto delle informazioni di autenticazione predefinito è impostato su **ADOPTCTX(YES)** e l'utente `fred` è collegato. Sono configurate le seguenti due regole CHLAUTH:

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

Il seguente comando viene emesso, con l'intenzione di autenticare il comando come contesto di sicurezza adottato dall'utente bob:

```
runmqsc -c -u bob QMGR
```

Infatti, il gestore code utilizza il contesto di sicurezza di `fred`, non bob, e la connessione non riesce.

Per utilizzare il contesto di sicurezza di bob, **ChlauthEarlyAdopt** deve essere impostato su Y.

PasswordProtection = Compatibile (predefinito) |always|facoltativo|warn

Le credenziali di autenticazione che le applicazioni IBM MQ client specificano quando si collegano a un gestore code possono essere protette utilizzando la funzione di protezione della password MQCSP IBM MQ, se la connessione non utilizza la codifica TLS.

La protezione della password MQCSP è utile per scopi di test e sviluppo in quanto l'utilizzo della protezione della password MQCSP è più semplice rispetto all'impostazione della crittografia TLS, ma non così sicuro.

Per ulteriori informazioni sulla protezione delle credenziali nella struttura MQCSP e sui valori che possono essere impostati per questo attributo, consultare [Protezione password MQCSP](#).

IgnoreSeqNumberMismatch = NO (predefinito) | YES

Gli MCA (Message Channel Agent) alle due estremità di un canale conservano il conteggio del numero di messaggi inviati attraverso il canale per mantenere la sincronizzazione. La sincronizzazione può essere persa, ad esempio se la definizione di canale ad un'estremità viene eliminata e quindi ricreata. In queste circostanze potrebbe essere richiesto un `RESET CHANNEL` per riconoscere che i dati di sincronizzazione sono stati persi e consentire al canale di continuare l'avvio.

L'attributo **IgnoreSeqNumberMismatch** deve essere impostato sul gestore code del destinatario.

Effettivamente, questo attributo esegue un comando di reimpostazione del canale sul canale ricevente.

Questo attributo controlla il modo in cui il gestore code gestisce una mancata corrispondenza del numero di sequenza durante l'avvio del canale utilizzando i seguenti valori:

No

I numeri di sequenza del canale vengono controllati durante la risincronizzazione del canale, se i due MCA non concordano sullo stesso numero di sequenza, verrà riportato il messaggio di errore AMQ9526 e il canale non verrà avviato.

sì

I numeri di sequenza del canale vengono controllati durante la risincronizzazione del canale, ma se i due MCA non concordano sullo stesso numero di sequenza, verrà notificato il messaggio di avviso AMQ9703 e l'avvio del canale continuerà. Questo valore di attributo non dovrebbe essere necessario in circostanze normali. Quando è noto che i dati di sincronizzazione sono stati persi, ad esempio durante il ripristino di emergenza, questa opzione evita di dover riconoscere manualmente ogni mancata corrispondenza del numero di sequenza. La specifica di questo valore

ha un effetto simile a quello di un amministratore che immette automaticamente un **RESET CHANNEL** in risposta a ogni mancata corrispondenza di numeri di sequenza.

ChlauthIgnoreUserCode = N (predefinito) | Y

Consente a un gestore code di rendere il nome utente corrispondente all'interno delle regole CHLAUTH non sensibili al maiuscolo / minuscolo. Questa opzione consente di:

- CLNTUSER nelle regole CHLAUTH TYPE (USERMAP) da mettere in corrispondenza senza distinzione tra maiuscole e minuscole
- USERLIST nelle regole CHLAUTH TYPE (BLOCKUSER) da mettere in corrispondenza in modo insensibile al maiuscolo / minuscolo

I valori validi per **ChlauthIgnoreUserCode** sono i seguenti:

N

Le regole di autenticazione del canale tentano di mettere in corrispondenza l'identificazione dell'utente client con la sensibilità al maiuscolo / minuscolo, ad esempio una regola che specifica CLNTUSER ('Fred') non corrisponderà a 'fred' o 'FRED', ma solo a un identificativo utente di 'Fred '. Questo è il valore predefinito.

Y

Le regole di autenticazione del canale tentano di mettere in corrispondenza l'identificazione utente del client con l'insensibilità al maiuscolo / minuscolo, ad esempio una regola di autenticazione del canale con TYPE (USERMAP) o TYPE (USERBLOCK) che specifica CLNTUSER ('Fred') corrisponderà a qualsiasi variazione di maiuscolo / minuscolo, ad esempio gli identificatori utente 'Fred', 'FRED' e 'fred ' corrispondono tutti.

Tenere presente che, quando si ignorano gli identificativi utente durante la corrispondenza delle regole di autenticazione del canale, è possibile che più di una regola corrisponda. Se ciò si verifica, la regola corrispondente non è definita. Ad esempio, con le seguenti regole, se l'utente 'fred' si connette a un gestore code tramite il canale CLIENT, potrebbe essere associato a 'mquser1' o 'mquser2':

```
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('fred') USERSRC(MAP) MCAUSER('mquser1')
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('FRED') USERSRC(MAP) MCAUSER('mquser2')
```

Per evitare qualsiasi incertezza quando si utilizza ChlauthIgnoreUserCode=Y, evitare di definire regole CHLAUTH che si sovrapporrebbero e risulterebbero in un comportamento diverso quando si utilizza una corrispondenza non sensibile al maiuscolo / minuscolo.

ChlauthIssueAvvertenza = y

Impostare questo attributo se si desidera che il messaggio AMQ9787 venga generato quando si imposta l'attributo WARN = YES sul comando **SET CHLAUTH**.

Stanza di esempio

```
Channels:
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD
  PipelineLength=2
```

Concetti correlati

“Stati del canale” a pagina 237

Un canale può essere in uno dei tanti stati in qualsiasi momento. Alcuni stati hanno anche sottostati. Da un determinato stato un canale può spostarsi in altri stati.

Multi Stanza di connessione del file qm.ini

La stanza Connection definisce il tipo di binding predefinito.

Utilizzare la stanza Connection nel file qm.ini per specificare il tipo di binding predefinito.

Linux

Windows

In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code IBM MQ Explorer Extended .

Nota: È necessario creare una stanza di connessione, se ne è necessaria una.

DefaultBindTipo = SHARED (valore predefinito) | ISOLATO

Se **DefaultBindType** è impostato su ISOLATO, le applicazioni e il gestore code vengono eseguiti in processi separati e non viene condivisa alcuna risorsa.

Se **DefaultBindType** è impostato su SHARED, le applicazioni e il gestore code vengono eseguiti in processi separati, ma alcune risorse vengono condivise tra loro.

Il valore predefinito è SHARED.



Attenzione: DefaultBindType si applica a tutte le chiamate MQCONN e a tutte le chiamate MQCONNX con MQCNO_STANDARD_BINDING.

La modifica di **DefaultBindType** potrebbe causare il peggioramento delle prestazioni di alcune applicazioni.

Stanza di esempio

```
Connection:  
DefaultBindType=SHARED
```

Multi


Registrazione dei messaggi diagnostici

I log dei messaggi diagnostici di IBM MQ sono un meccanismo che consente ai vari componenti del sistema IBM MQ di notificare i messaggi diagnostici relativi alle modifiche e ai problemi di configurazione e di stato di runtime di IBM MQ .

Questi log sono a volte indicati come IBM MQ *log di errore*, ma contengono sempre IBM MQ informazioni e messaggi di avvertenza, nonché messaggi di errore. I tre componenti principali di IBM MQ che riportano a questi log sono:

- Gestori code
- IBM MQ Client
- Il resto del sistema IBM MQ

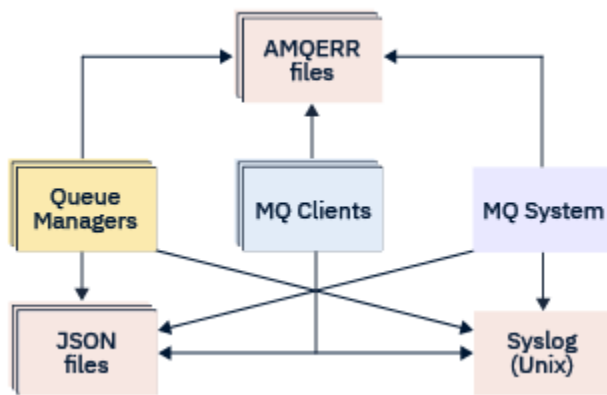
IBM MQ supporta la creazione di report dei messaggi di diagnostica tramite diversi metodi noti come *servizi di messaggi di diagnostica*, consentendo un approccio personalizzato per la registrazione e l'utilizzo di tali informazioni:

- File di log AMQERRnn
- File di log formattati JSON
-  Syslog in formato JSON

L'output JSON di IBM MQ è formattato come oggetti JSON a riga singola, in modo che ogni singola riga del log JSON o del record Syslog, rappresenti un oggetto JSON valido. L'intero log non è incapsulato come un singolo oggetto JSON.

La seguente figura mostra che i gestori code, i client IBM MQ e il sistema IBM MQ possono *tutti* i messaggi di diagnostica utilizzando i metodi descritti.

Figura 5. Come le diverse parti di IBM MQ possono riportare i messaggi diagnostici



Modalità di configurazione dei log di diagnostica IBM MQ :

I log di diagnostica sono definiti e personalizzati utilizzando le stanze all'interno del file `qm.ini` particolare del componente IBM MQ che li richiede. Ogni endpoint di registrazione univoco è definito sotto la propria intestazione di stanza all'interno del file ini, insieme a tutte le personalizzazioni definite al suo interno. Le personalizzazioni possono includere:

- La dimensione dei file di log su cui eseguire il wrap, prima del rollover; non applicabile a Syslog
- Qualsiasi filtro basato sulla severità dei messaggi di log e
- Qualsiasi codice di messaggio specifico da eliminare.

IBM MQ può essere configurato per scrivere in uno o in tutti e tre i tipi di endpoint di registrazione, consentendo a particolari stanze di log di soddisfare particolari ruoli. Allo stesso modo, è possibile definire più servizi file. Ad esempio:

- Il formato JSON semplifica l'analisi tramite strumenti automatizzati in ambienti locali e cloud.
- L'output syslog consente ai componenti IBM MQ di integrare le informazioni di diagnostica in un'ubicazione di registrazione del SO comune in linea con altri prodotti sul sistema.
- Gli endpoint di log filtrati in base alla severità, consentendo a determinati file di log di registrare, ad esempio, solo errori gravi nel sistema.

Indipendentemente dallo stile di registrazione della diagnostica configurato, i file di diagnostica tradizionali contenuti nella directory di log del sistema IBM MQ (`/var/mqm/errors/AMQERRnn.log`) e nella directory di log del gestore code specifico (`/var/mqm/qmgrs/<qmgr_name>/errors/AMQERRnn.log`) vengono sempre scritti, in aggiunta a qualsiasi altra configurazione di registrazione utilizzata.

Solo per i gestori code, la configurazione facoltativa di questi log obbligatori può essere eseguita specificando attributi di [“Stanze del servizio messaggi diagnostici”](#) a pagina 134.

Aree di stanza differenti

Le stanze aggiuntive possono essere applicate a diverse aree di IBM MQ.

Gestore code (`qm.ini`)

Si applica al messaggio di log generato dal gestore code

Sistema (`mqs.ini`)

Si applica ai messaggi di log generati dal sistema. Questa opzione non è specifica per un gestore code, tranne quando un gestore code non può accedere o scrivere nei propri log.

Modelli (`mqs.ini`)

Una o più stanze come modelli, che vengono copiate in `qm.ini` quando viene creato un gestore code.

Client (`mqclient.ini`)

Si applica all'operazione client, ad esempio `runmqsc` in modalità client per un gestore code remoto.

Conversione tra i log formattati JSON e quelli formattati in modo tradizionale

Il comando `mqrcc` è stato migliorato per consentire un numero di conversioni tra JSON e i log formattati in modo tradizionale e tra lingue differenti.

Riferimenti correlati

“Stanze del servizio messaggi diagnostici” a pagina 134

Le opzioni del servizio messaggi di diagnostica disponibili abilitano la personalizzazione della registrazione della diagnostica IBM MQ, in modo che l'output del log possa essere indirizzato a diversi endpoint di log da diversi componenti di IBM MQ.

“stanza `QMErrorLog`” a pagina 133

Utilizzare la stanza del log degli errori del gestore code `QMErrorLog` nel file `qm.ini` per adattare l'operazione e il contenuto dei log degli errori IBM MQ.

“Servizi di messaggi diagnostici” a pagina 137



È possibile definire i seguenti servizi di messaggi diagnostici e i relativi attributi specifici del servizio, specificati nelle sezioni `DiagnosticSystemMessages`, `DiagnosticMessages` e `DiagnosticMessagesTemplate` dei file di configurazione:

stanza `QMErrorLog`

Utilizzare la stanza del log degli errori del gestore code `QMErrorLog` nel file `qm.ini` per adattare l'operazione e il contenuto dei log degli errori IBM MQ.


Il servizio `QMErrorLog` è il servizio di log di diagnostica IBM MQ tradizionale utilizzato per emettere messaggi di diagnostica relativi al gestore code. Il servizio `QMErrorLog` viene eseguito continuamente e non può essere disattivato, ma può essere personalizzato in una certa misura.

È possibile utilizzare la sezione `QMErrorLog` del file `qm.ini` per escludere alcuni messaggi dalla scrittura nel log degli errori del gestore code. È inoltre possibile impedire che i messaggi vengano scritti nel log degli errori per un determinato periodo di tempo.

  In alternativa, anziché modificare direttamente il file `qm.ini`, è possibile utilizzare la pagina delle proprietà del gestore code esteso in IBM MQ Explorer per escludere ed eliminare i messaggi con gli attributi **Messaggi esclusi**, **Messaggi soppressi** e **Intervallo messaggi soppressi**.



Attenzione:

-  È possibile utilizzare IBM MQ Explorer per apportare le modifiche solo se si sta utilizzando un gestore code locale sulla piattaforma Windows.
- La stanza `QMErrorLog` non è applicabile al file di configurazione del sistema IBM MQ, `mqs.ini`, o al file di configurazione del client, generalmente denominato `mqclicent.ini`.

I seguenti attributi possono essere inclusi nella stanza `QMErrorLog`:

ErrorLogDimensione = *maxsize*

Specifica la dimensione del log degli errori del gestore code copiato nel backup. *maxsize* deve essere compreso tra 32768 e 2147483648 byte. Se **ErrorLogSize** non viene specificato, viene utilizzato il valore predefinito di 33554432 byte (32 MB).

È possibile utilizzare questo attributo per ridurre la dimensione massima al valore massimo precedente di 2 MB, se richiesto.

È possibile impostare la dimensione del log utilizzando la variabile di ambiente **MQMAXERRORLOGSIZE**.

ExcludeMessage= *msgIds*

Specifica i messaggi che non devono essere scritti nel log degli errori del gestore code.

Per ulteriori informazioni, consultare `ExcludeMessage` in “Stanze del servizio messaggi diagnostici” a pagina 134.

SuppressMessage= msgIds

Specifica i messaggi scritti nel log degli errori del gestore code una sola volta in un intervallo di tempo specificato. Se lo stesso ID messaggio viene specificato in SuppressMessage e ExcludeMessage, il messaggio viene escluso.

Questa opzione non è applicabile ai servizi di messaggi di diagnostica definiti in mqclient.ini. Per ulteriori informazioni, consultare SuppressMessage in “Stanze del servizio messaggi diagnostici” a pagina 134.

SuppressInterval= lunghezza

Specifica l'intervallo di tempo, in secondi, in base al quale i messaggi specificati in SuppressMessage vengono scritti nel log degli errori del gestore code una sola volta. *length* deve essere compreso tra 1 e 86400 secondi. Se SuppressInterval non viene specificato, viene utilizzato il valore predefinito di 30 secondi.

Stanza di esempio

```
QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30
```

Concetti correlati

“File di configurazione del gestore code, qm.ini” a pagina 107

Un file di configurazione del gestore code, qm.ini, contiene informazioni relative a uno specifico gestore code. Gli attributi che è possibile utilizzare per modificare la configurazione di un singolo gestore code sovrascrivono le impostazioni per IBM MQ.

Riferimenti correlati

“Stanze del servizio messaggi diagnostici” a pagina 134

Le opzioni del servizio messaggi di diagnostica disponibili abilitano la personalizzazione della registrazione della diagnostica IBM MQ, in modo che l'output del log possa essere indirizzato a diversi endpoint di log da diversi componenti di IBM MQ.

Multi Stanze del servizio messaggi diagnostici

Le opzioni del servizio messaggi di diagnostica disponibili abilitano la personalizzazione della registrazione della diagnostica IBM MQ, in modo che l'output del log possa essere indirizzato a diversi endpoint di log da diversi componenti di IBM MQ.

Abilitare ulteriori servizi di messaggi diagnostici, utilizzando una stanza con uno dei seguenti nomi:

- **DiagnosticSystemMessages**

Definisce i servizi utilizzati quando viene generato un messaggio diagnostico che va alla registrazione errori di sistema. Valido nei file mqs.ini o mqclient.ini.

Le applicazioni client utilizzano una sezione **DiagnosticSystemMessages** nel file mqclient.ini e in mqs.ini, la stanza **DiagnosticSystemMessages** controlla i messaggi per un'applicazione server che non dispone di un contesto gestore code.

È possibile configurare un gestore code e le applicazioni che scrivono ulteriormente tutti i messaggi nel servizio syslog.

- **DiagnosticMessages**

Definisce i servizi utilizzati quando viene generato un messaggio diagnostico che va al log degli errori del gestore code. Valido solo nel file qm.ini.

- **DiagnosticMessagesTemplate**

Una stanza copiata dal file mqs.ini in **DiagnosticMessages** nel file qm.ini quando viene creato un gestore code.

Per visualizzare messaggi diagnostici, utilizzare il comando `mqrcl`.

Attributi delle stanze



Attenzione: Servizio e un nome di stanza sono obbligatori.

nome= < nomestanza >

Nome di una stanza. Il valore deve essere univoco in un file ini.

Servizio = tipo di servizio

Questo attributo definisce un servizio, in cui il nome del servizio non è sensibile al maiuscolo / minuscolo, abilitato da questa stanza.

Ad esempio, per abilitare `syslog` come servizio aggiuntivo, immettere quanto segue:

```
Service=syslog
```


Consultare “[Servizi di messaggi diagnostici](#)” a pagina 137 e i relativi attributi specifici disponibili per l'utilizzo con le stanze del servizio messaggi diagnostici.

È possibile aggiungere i seguenti attributi facoltativi alle stanze:

- [ExcludeMessage](#)
- [SuppressMessage](#)
- [SuppressInterval](#)
- “Gravità” a pagina 136

ExcludeMessage= msgIds

Specifica i messaggi che non devono essere scritti nel log degli errori del gestore code. Se il sistema IBM MQ è molto utilizzato, con molti canali in fase di arresto e avvio, un numero elevato di messaggi informativi viene inviato alla console z/OS e al log della copia cartacea. Il bridge IBM MQ - IMS e il gestore buffer potrebbero anche produrre un numero elevato di messaggi informativi, pertanto l'esclusione dei messaggi impedisce di ricevere un numero elevato di messaggi, se necessario. `msgIds` contiene un elenco separato da virgole di ID messaggio provenienti da:

- 5211 - È stata superata la lunghezza massima del nome proprietà.
- 5973 - Sottoscrizione di pubblicazione / sottoscrizione distribuita non abilitata
- 5974 - Pubblicazione di pubblicazione / sottoscrizione distribuita non abilitata
- 6254 - Il sistema non è stato in grado di caricare dinamicamente la libreria condivisa
-  7163 - Messaggio avviato dal lavoro (solo IBM i)
- 7234 - Numero di messaggi caricati
- 8245 - L'entità non dispone di autorizzazione sufficiente per visualizzare l'oggetto
- 9001 - Programma del canale terminato normalmente
- 9002 - Programma del canale avviato
- 9202 - Host remoto non disponibile
- 9208 - Errore di ricezione dall'host
- 9209 - Connessione chiusa
- 9228 - Impossibile avviare il responder del canale
- 9489 - Limite massimo di istanze SVRCONN superato
- 9490 - Limite massimo di istanze SVRCONN per client superato
- 9508 - Impossibile connettersi al gestore code
- 9524 - Gestore code remoto non disponibile
- 9528 - Chiusura del canale richiesta dall'utente
- 9545 - Intervallo di disconnessione scaduto
- 9558 - Canale remoto non disponibile
- 9637 - Al canale manca un certificato

- 9776 - Il canale è stato bloccato dall'ID utente
- 9777 - Il canale è stato bloccato dalla mappa NOACCESS
- 9782 - La connessione è stata bloccata dall'indirizzo
- 9999 - Programma del canale terminato in modo anomalo

SuppressMessage= msgIds

Specifica i messaggi scritti nel log degli errori del gestore code una sola volta in un intervallo di tempo specificato. Se il sistema IBM MQ è molto utilizzato, con molti canali in fase di arresto e avvio, un numero elevato di messaggi informativi viene inviato alla console z/OS e al log della copia cartacea. Il bridge IBM MQ - IMS e il gestore buffer potrebbero anche produrre un numero elevato di messaggi informativi, pertanto l'eliminazione dei messaggi impedisce di ricevere un certo numero di messaggi ripetuti, se necessario. L'intervallo di tempo è specificato da `SuppressInterval`. `msgIds` contiene un elenco separato da virgole di identificatori di messaggi dai seguenti:

- 5211 - È stata superata la lunghezza massima del nome proprietà.
- 5973 - Sottoscrizione di pubblicazione / sottoscrizione distribuita non abilitata
- 5974 - Pubblicazione di pubblicazione / sottoscrizione distribuita non abilitata
- 6254 - Il sistema non è stato in grado di caricare dinamicamente la libreria condivisa
- IBM i 7163 - Messaggio avviato dal lavoro (solo IBM i)
- 7234 - Numero di messaggi caricati
- 8245 - L'entità non dispone di autorizzazione sufficiente per visualizzare l'oggetto
- 9001 - Programma del canale terminato normalmente
- 9002 - Programma del canale avviato
- 9202 - Host remoto non disponibile
- 9208 - Errore di ricezione dall'host
- 9209 - Connessione chiusa
- 9228 - Impossibile avviare il responder del canale
- 9489 - Limite massimo di istanze SVRCONN superato
- 9490 - Limite massimo di istanze SVRCONN per client superato
- 9508 - Impossibile connettersi al gestore code
- 9524 - Gestore code remoto non disponibile
- 9528 - Chiusura del canale richiesta dall'utente
- 9545 - Intervallo di disconnessione scaduto
- 9558 - Canale remoto non disponibile
- 9637 - Al canale manca un certificato
- 9776 - Il canale è stato bloccato dall'ID utente
- 9777 - Il canale è stato bloccato dalla mappa NOACCESS
- 9782 - La connessione è stata bloccata dall'indirizzo
- 9999 - Programma del canale terminato in modo anomalo

Se lo stesso ID messaggio viene specificato in `SuppressMessage` e `ExcludeMessage`, il messaggio viene escluso.

Questa opzione non è applicabile ai servizi di messaggi di diagnostica definiti in `MQ client.ini`.

SuppressInterval= lunghezza

Specifica l'intervallo di tempo, in secondi, in cui i messaggi specificati in `SuppressMessage` vengono scritti una sola volta nel log degli errori del gestore code. `length` deve essere compreso tra 1 e 86400 secondi. Se `SuppressInterval` non viene specificato, viene utilizzato il valore predefinito di 30 secondi.

Gravità

Un elenco separato da virgole di livelli di severità, in cui il nome del livello di severità non è sensibile al maiuscolo / minuscolo. I valori consentiti sono:

- I (o Informazioni o 0)
- W (o Avvertenza o 10)

- E (o Errore o 20 e 30)
- S (o Stop o 40)
- T (o Sistema o 50)

Note:

1. Il valore predefinito è: a11
2. Solo i messaggi nei livelli di severità selezionati vengono presentati al servizio.

In alternativa, è possibile utilizzare il carattere più (+) che visualizza il livello di errore specificato e tutti i livelli superiori. Ad esempio, per visualizzare tutti gli errori:

```
Severities=E+
```

Riferimenti correlati

[“stanza QMErrorLog” a pagina 133](#)

Utilizzare la stanza del log degli errori del gestore code QMErrorLog nel file `qm.ini` per adattare l'operazione e il contenuto dei log degli errori IBM MQ .

[“Servizi di messaggi diagnostici” a pagina 137](#)

È possibile definire i seguenti servizi di messaggi diagnostici e i relativi attributi specifici del servizio, specificati nelle sezioni `DiagnosticSystemMessages`, `DiagnosticMessages` e `DiagnosticMessagesTemplate` dei file di configurazione:

Multi *Servizi di messaggi diagnostici*

È possibile definire i seguenti servizi di messaggi diagnostici e i relativi attributi specifici del servizio, specificati nelle sezioni `DiagnosticSystemMessages`, `DiagnosticMessages` e `DiagnosticMessagesTemplate` dei file di configurazione:

Sono definiti i seguenti servizi di messaggi diagnostici:

File

Questo servizio invia i messaggi non filtrati a un file in modo simile al servizio QMErrorLog . Il formato testuale esistente o il formato JSON specificato viene utilizzato in base al **Format** specificato. Per impostazione predefinita, esistono tre file denominati `AMQERR01.LOG`, `AMQERR02.LOG` e `AMQERR03.LOG` o `AMQERR01.json`, `AMQERR02.json` e `AMQERR03.json`, a seconda della proprietà **Format** e questi rollover si basano sulla dimensione configurata.

I seguenti attributi sono supportati solo in una stanza File:

FilePath

Il percorso in cui vengono scritti i file di log. Il valore predefinito è lo stesso percorso dei file `AMQERR01.log` , ossia il sistema o il gestore code. Il percorso deve essere assoluto, ma può includere inserimenti sostituibili. Ad esempio:

+ MQ_Q_MGR_DATA_PATH +

Il percorso completo dell'elemento principale della directory dei messaggi di diagnostica del gestore code. I valori predefiniti sono:

- **Linux** / **AIX** Su piattaforme AIX and Linux : `/var/mqm/qmgrs/<QM_name>`
- **Windows** su Windows, `C:\Program Data\IBM\MQ\qmgrs\<QM_name>`

+ MQ_DATA_PATH +

Il percorso completo della directory dei messaggi di diagnostica del sistema. I valori predefiniti sono:

- **Linux** / **AIX** Su piattaforme AIX and Linux : `/var/mqm`
- **Windows** Su Windows: `C:\Program Data\IBM\MQ`

È necessario creare questo percorso con le autorizzazioni appropriate, se non si utilizza la directory degli errori esistente.

FilePrefix

Il prefisso dei file di log. Il valore predefinito è AMQERR.

FileSize

La dimensione alla quale il log esegue il rollover. Il valore predefinito è 32MB, come con la proprietà **ErrorLogSize** di “stanza QMErrorLog” a pagina 133, che è semanticamente identica.

Nota: La proprietà **ErrorLogSize** si applica solo al servizio di registrazione errori predefinito, non ai servizi di diagnostica personalizzati.

È possibile impostare la dimensione del log utilizzando la variabile di ambiente **MQMAXERRORLOGSIZE**.

Format

Il formato del file. Il valore può essere *text* (per ulteriori servizi di stile QMErrorLog) o *json*, che è il valore predefinito.

Il suffisso del file è .LOG o .json in base all'impostazione di questo attributo.

Ad esempio, modificare il file `qm.ini` del gestore code e aggiungere la stanza seguente:

```
DiagnosticMessages:
  Service = File
  Name = JSONLogs
  Format = json
  FilePrefix = AMQERR
```

Dopo il riavvio, il gestore code avrà i file `AMQERR0x.json` nella directory `ERRORS`.

È possibile definire più servizi File. Ciò consente la configurazione, come mostrato nei seguenti esempi, in cui i messaggi di tag differenti sono suddivisi in diverse serie di log:

```
DiagnosticMessages:
  Name=ErrorsToFile
  Service=File
  Severities=E+
  FilePrefix=OnlyErrors

DiagnosticMessages:
  Name=NonErrorstoFile
  Service=File
  Severities=1 W
  FilePrefix=Information
```

Linux

AIX

Registro eventi sistema

Il servizio Syslog non è disponibile su Windows o IBM i

È possibile definire solo un servizio Syslog e il servizio Syslog invia tutti i messaggi non filtrati a syslog utilizzando la specifica dei messaggi diagnostici formato JSON. Le informazioni vengono aggiunte a syslog nell'ordine mostrato nella tabella, iniziando con `msgID` e inserimenti.

La severità del messaggio è associata al livello syslog nel seguente modo:

Gravità	Livello
0	LOG_INFO
10	LOG_AVVERTENZA
20	ERR LOGO
30	ERR LOGO
40	LOG_ALERT
50	LOG_ALERT

Il seguente attributo è supportato solo in una stanza syslog:

Ident

Definisce il valore **ident** associato alle entrate syslog. Il valore predefinito è *ibm - mq*.

Il seguente esempio mostra i messaggi di errore inviati al Syslog:

```
DiagnosticMessages:  
Name=ErrorsToSyslog  
Ident=mq  
Service=Syslog  
Severities=E+
```

Consultare [“Stanze del servizio messaggi diagnostici”](#) a pagina 134 per ulteriori informazioni sugli attributi generici della stanza.

Note:

1. Solo per il servizio File, è possibile avere più stanze, ognuna con un nome diverso. Solo la definizione, utilizzando il nome finale nella sequenza, diventa effettiva.
2. Le modifiche al valore di una stanza diventano effettive solo quando il gestore code viene riavviato.

Multi Stanza ExitPath del file qm.ini

La stanza ExitPath specifica il percorso per i programmi di uscita utente sul sistema gestore code.

Utilizzare la stanza ExitPath nel file `qm.ini` per specificare il percorso per i programmi di uscita utente sul sistema del gestore code.

Linux **Windows** In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code IBM MQ Explorer Exits .

ExitsDefaultPercorso = *stringa*

L'attributo ExitsDefaultPath specifica l'ubicazione di:

- Uscite di canale a 32 bit per i client
- Uscite di canali a 32 bit e uscite di conversione dati per server
- File di caricamento switch XA non qualificati

ExitsDefaultPath64= *stringa*

L'attributo ExitsDefaultPath64 specifica l'ubicazione di:

- Uscite di canale a 64 bit per client
- Uscite di canali a 64 bit e uscite di conversione dati per server
- File di caricamento switch XA non qualificati

Stanza di esempio

```
ExitPath:  
ExitsDefaultPath=/var/mqm/exits  
ExitsDefaultPath64=/var/mqm/exits64
```

Multi ExitPropertiesStanza locale del file qm.ini

La stanza ExitPropertiesLocal specifica le informazioni sulle proprietà di uscita su un gestore code.

Utilizzare la stanza ExitPropertiesLocal nel file `qm.ini` per specificare le informazioni sulle proprietà di uscita su un gestore code.

Linux **Windows** In alternativa, in Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code del cluster IBM MQ Explorer .

Windows In alternativa, su Windows è possibile specificare queste informazioni utilizzando il comando **amqmdain** .

Per impostazione predefinita, questa impostazione viene ereditata dall'attributo **CLWLMode** nella stanza ExitProperties della configurazione della macchina (descritta in [“Stanza ExitProperties del file mqs.ini”](#) a pagina 102). Modificare questa impostazione solo se si desidera configurare questo gestore code in un modo diverso. Questo valore può essere sovrascritto per i gestori code individuali utilizzando l'attributo della modalità del workload del cluster nella pagina delle proprietà del gestore code del cluster.

Utilizzare la stanza ExitProperties nel file mqs . ini per specificare opzioni di configurazione utilizzate dai programmi di uscita del gestore code.

Linux **Windows** In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà di IBM MQ Explorer Extended IBM MQ .

CLWLMode = SAFE (predefinito) | FAST

L'uscita del carico di lavoro cluster (CLWL) consente di specificare quale coda cluster nel cluster aprire in risposta a una chiamata MQI (ad esempio, MQOPEN, MQPUT). L'uscita CLWL viene eseguita in modalità FAST o SAFE in base al valore specificato nell'attributo **CLWLMode** . Se si omette l'attributo **CLWLMode** , l'uscita del carico di lavoro cluster viene eseguita in modalità SAFE.

SICURA

Eseguire l'uscita CLWL in un processo separato dal gestore code. Questa è l'opzione predefinita.

Se si verifica un problema con l'uscita CLWL scritta dall'utente durante l'esecuzione in modalità SAFE, si verifica quanto segue:

- Il processo del server CLWL (amqzlw0) ha esito negativo.
- Il gestore code riavvia il processo server CLWL.
- L'errore viene riportato nel log degli errori. Se è in corso una chiamata MQI, si riceve una notifica sotto forma di codice di ritorno.

L'integrità del gestore code viene preservata.

Nota: L'esecuzione dell'uscita CLWL in un processo separato potrebbe influire sulle prestazioni.

VELOCE

Eseguire l'uscita cluster in linea nel processo del gestore code.

Specificando questa opzione si migliorano le prestazioni evitando i costi di commutazione del processo associati all'esecuzione in modalità SAFE, ma a discapito dell'integrità del gestore code. Si consiglia di eseguire l'uscita CLWL in modalità FAST solo se si è certi che non vi sono problemi con l'uscita CLWL e si è particolarmente preoccupati per le prestazioni.

Se si verifica un problema quando l'uscita CLWL è in esecuzione in modalità FAST, il gestore code avrà esito negativo e si corre il rischio che l'integrità del gestore code venga compromessa.

Stanza di esempio

```
ExitPropertiesLocal:  
CLWLMode=SAFE
```

IBM i **Linux** **AIX** **Stanza del file system del file qm.ini**

La stanza File system specifica se le autorizzazioni impostate sui log degli errori del gestore code devono rimanere invariate o devono essere nuovamente modificate ai valori predefiniti.

Si prevede che le autorizzazioni predefinite impostate sui file di log degli errori siano utili nella maggior parte dei casi e, pertanto, non è necessario che la maggior parte degli amministratori IBM MQ le modifichino.

Tuttavia, l'amministratore IBM MQ potrebbe voler modificare le autorizzazioni sui propri file di log degli errori, nel qual caso dovrebbe impostare l'opzione della stanza `Filesystem` **ValidateAuth=No**, che fa in modo che il gestore code lasci le autorizzazioni inalterate in seguito.

Il funzionamento predefinito (senza **ValidateAuth=No**) è che il gestore code controlla le autorizzazioni file dei log degli errori del gestore code e le modifica di nuovo ai valori predefiniti. Questo controllo può verificarsi in qualsiasi momento, anche durante un'operazione di fine o avvio del gestore code.

Stanza di esempio

```
Filesystem:
  ValidateAuth=No
```

V 9.4.0 Linux AIX Stanza JWKS del file mq.ini

Utilizzare la sezione **JWKS** per informare il gestore code su come richiamare una serie di chiavi pubbliche che possono convalidare i token firmati da questo emittente.

EndPoint=stringa

Specifica l'URL per un documento JWKS che fornisce le chiavi pubbliche richieste per convalidare i token firmati da questo emittente.

Questo server specificato deve fornire una connessione sicura (HTTPS) per richiamare le chiavi. Il gestore code deve essere stato configurato anche per stabilire connessioni HTTP protette, consultare [HTTPSKeyStore](#).

Se non si è certi del valore corretto per questo attributo, consultare l'amministratore del server di autenticazione.

IssuerName=stringa

Deve essere il **IssuerName** presente nei token di autenticazione JWT forniti / firmati da questo emittente.

È possibile stabilire questa situazione esaminando la documentazione per il server di autenticazione o esaminando un JWT di esempio emesso dal servizio.

UserClaim=stringa

Fornisce il nome di un'asserzione (coppia chiave / valore all'interno di un token di autenticazione) che il gestore code utilizzerà come ID utente IBM MQ quando stabilisce le autorizzazioni IBM MQ.

Se i token da questo endpoint vengono utilizzati solo per autenticazione, ossia **ADOPTCTX** è NO, questo attributo è facoltativo, altrimenti è obbligatorio.

Stanza di esempio

```
JWKS:
  Endpoint=https://myauthserver.example/jwks
  IssuerName=https://myauthserver.example/jwks
  UserClaim=MQUser
```


Informazioni correlate

[Configurazione di un gestore code per accettare i token di autenticazione utilizzando un endpoint JWKS](#)

Multi Stanza di log del file qm.ini

La stanza Log specifica le informazioni sulla registrazione su un gestore code.

Utilizzare la stanza Log nel file `qm.ini` per specificare informazioni sulla registrazione su un gestore code.

 In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà IBM MQ Explorer **Gestore code log**.

Per impostazione predefinita, queste impostazioni vengono ereditate dalle impostazioni specificate per le impostazioni di log predefinite del gestore code (descritte in [“Stanza LogDefaults del file mqs.ini”](#) a pagina 102). Modificare queste impostazioni solo se si desidera configurare questo gestore code in un modo diverso.

Per ulteriori informazioni sul calcolo delle dimensioni dei log, consultare [“Calcolo della dimensione del log”](#) a pagina 671.

Nota: I limiti forniti nel seguente elenco di parametri sono impostati da IBM MQ. I limiti del sistema operativo potrebbero ridurre la dimensione massima possibile del log.

LogPrimaryFiles = 3 (valore predefinito) | 2-254 (Windows) | 2-510 (sistemi AIX and Linux)

I file di log assegnati quando viene creato il gestore code.

Il numero minimo di file di log primari che è possibile avere è 2 e il massimo è 254 su Windows e 510 su sistemi AIX and Linux . Il valore predefinito è 3.

Il numero totale di file di log primari e secondari non deve superare 255 su sistemi Windows e 511 su sistemi AIX and Linux e non deve essere inferiore a 3.

Una volta creato o avviato il gestore code, il valore viene configurato automaticamente. È possibile modificarlo una volta creato il gestore code. Tuttavia, una modifica del valore non è effettiva fino a quando il gestore code non viene riavviato e l'effetto potrebbe non essere immediato.

LogSecondaryFiles = 2 (predefinito) | 1-253 (Windows) | 1-509 (sistemi AIX and Linux)

I file di log assegnati quando i file primari sono esauriti.

Il numero minimo di file di log secondari è 1 e il massimo è 253 su Windows e 509 su sistemi AIX and Linux . Il numero predefinito è 2.

Il numero totale di file di log primari e secondari non deve superare 255 su sistemi Windows e 511 su sistemi AIX and Linux e non deve essere inferiore a 3.

Il valore viene esaminato quando il gestore code viene avviato. È possibile modificare questo valore, ma le modifiche non diventano effettive fino a quando il gestore code non viene riavviato e anche in questo caso l'effetto potrebbe non essere immediato.

LogFilePagine = numero

I dati di log sono contenuti in una serie di file denominati file di log. La dimensione del file di log è specificata in unità di pagine da 4 KB.

Il numero predefinito di pagine del file di log è 4096, fornendo una dimensione del file di log di 16 MB.

Su sistemi AIX and Linux , il numero minimo di pagine del file di log è 64 e su Windows il numero minimo di pagine del file di log è 32; in entrambi i casi, il numero massimo è 65 535.

Nota: La dimensione dei file di log specificati durante la creazione del gestore code non può essere modificata per un gestore code.

LogType = CIRCULAR (valore predefinito) | LINEAR| REPLICATED

Il tipo di registrazione che deve essere utilizzato dal gestore code. Il valore predefinito è CIRCULAR. Per ulteriori informazioni relative alla creazione di un gestore code con il tipo di registrazione richiesto, consultare la descrizione dell'attributo **LogType** in [“Stanza LogDefaults del file mqs.ini”](#) a pagina 102.

CIRCOLARE

Avviare il ripristino del riavvio utilizzando il log per eseguire il rollback delle transazioni in corso all'arresto del sistema.

Consultare [“Tipi di registrazione”](#) a pagina 665 per una spiegazione più completa della registrazione circolare.

LINEARE

Sia per il ripristino del riavvio che per il ripristino del supporto o dell'inoltro (creazione di dati persi o danneggiati riproducendo il contenuto della registrazione).

Consultare [“Tipi di registrazione” a pagina 665](#) per una spiegazione più completa della registrazione lineare.

CP4I **Replicata**

Utilizzato da un gruppo HA nativo per replicare i dati di log dall'istanza attiva alle istanze di replica.

Consultare [“Tipi di registrazione” a pagina 665](#) per una spiegazione più completa della registrazione replicata.

Nota: Il **LogType** di un gestore code non può essere modificato modificando questo attributo nel file `qm.ini`. Per modificare il **LogType** di un gestore code, è necessario utilizzare il comando **migmqlog**.

LogBufferPages=0 (predefinito) | 0 - 4096

La quantità di memoria assegnata ai record buffer per la scrittura, specificando la dimensione dei buffer in unità di pagine da 4 KB.

Il numero minimo di pagine di buffer è 18 e il massimo è 4096. Buffer più grandi portano ad una maggiore velocità di trasmissione, specialmente per messaggi più grandi.

Se si specifica 0 (valore predefinito), il gestore code seleziona la dimensione.




Se si specifica un numero compreso tra 1 e 17, il valore predefinito del gestore code è 18 (72 KB). Se si specifica un valore compreso tra 18 e 4096, il gestore code utilizza il numero specificato per impostare la quantità di memoria assegnata.

Il valore viene esaminato quando il gestore code viene avviato. Il valore può essere aumentato o diminuito entro i limiti indicati. Tuttavia, una modifica del valore non sarà effettiva fino al successivo avvio del gestore code.

LogPath= nome_directory



La directory in cui risiedono i file di log per un gestore code. Deve esistere su una periferica locale su cui il gestore code può scrivere e, preferibilmente, su un'unità diversa dalle code messaggi. La specifica di un'unità differente fornisce una protezione aggiuntiva in caso di errore del sistema.

Il valore predefinito è:

-  `C:\ProgramData\IBM\MQ\log` in Windows.
-   `/var/mqm/log` in sistemi AIX and Linux.

È possibile specificare il nome di una directory nel comando **crtmqm** utilizzando l'indicatore **-ld**. Quando un gestore code viene creato, viene creata anche una directory nella directory del gestore code, utilizzata per conservare i file di log. Il nome di questa directory è basato sul nome gestore code. Ciò garantisce che il percorso del file di log sia univoco e che sia conforme alle eventuali limitazioni sulle lunghezze dei nomi di directory.

Se non si specifica **-ld** sul comando **crtmqm**, viene utilizzato il valore dell'attributo **LogDefaultPath**.

  Sui sistemi AIX and Linux, l'ID utente `mqm` e il gruppo `mqm` devono avere autorizzazioni complete per i file di log. Se si modificano le ubicazioni di questi file, è necessario fornire personalmente tali autorizzazioni. Ciò non è richiesto se i file di log si trovano nelle ubicazioni predefinite fornite con il prodotto.

LogWriteIntegrity =SingleWrite|DoubleWrite|TripleWrite (predefinito)

Il metodo utilizzato dal programma di registrazione per scrivere in modo affidabile i record di log.

TripleWrite (predefinito)

Nota: è possibile selezionare `DoubleWrite` ma, in tal caso, il sistema l'interpreta come `TripleWrite`.

SingleWrite

Si consiglia di utilizzare SingleWrite, solo se il file - system e il dispositivo che ospita il log di recupero IBM MQ garantiscono esplicitamente l'atomicità delle scritture 4KB .

Ossia, quando una scrittura di una pagina di 4KB non riesce per un qualsiasi motivo, i soli due stati possibili sono la pre-immagine o la post-immagine. Non deve essere possibile alcuno stato intermedio.

Nota: Se è presente una simultaneità sufficiente nel tuo carico di lavoro persistente, c'è un vantaggio potenziale minimo nell'impostazione di un valore diverso da quello predefinito, TripleWrite.

Per ulteriori informazioni, consultare [“LogWriteIntegrity - utilizzando SingleWrite o TripleWrite” a pagina 144.](#)

LogManagement = Manuale (predefinito) | Automatico | Archivio

Il metodo utilizzato per gestire le estensioni di log, manualmente o dal gestore code. Il valore predefinito è Manuale.

L'attributo si applica solo quando **LogType** è LINEAR.

Se si modifica il valore **LogManagement**, la modifica non ha effetto fino al riavvio del gestore code.

Se viene trovato un valore non riconosciuto per l'attributo, il gestore code non verrà avviato fino a quando il valore non viene corretto.

 La proprietà **LogManagement** non è valida su IBM i.

Manuale (predefinito)

Le estensioni di log vengono gestite manualmente. Specificando questa opzione il gestore code non riutilizza o elimina le estensioni log, anche quando non sono più necessarie per il ripristino.

Automatico

Le estensioni log vengono gestite automaticamente dal gestore code. Specificando questa opzione il gestore code può riutilizzare o eliminare le estensioni log quando non sono più necessarie per il ripristino. Nessuna agevolazione viene effettuata per l'archiviazione.

Archivio

Le estensioni log sono gestite dal gestore code, ma è necessario notificare al gestore code quando l'archiviazione di ciascuna estensione log è completa.

Specificando questa opzione il gestore code è libero di riutilizzare o eliminare un'estensione log esistente, come viene notificato che un'estensione log non è più necessaria per il ripristino essa viene archiviata.

Eseguire questa notifica utilizzando il comando **RESET QMGR MQSC** o il comando PCF [Reimposta gestore code](#) .

Stanza di esempio

```
Log:
LogPrimaryFiles=3
LogSecondaryFiles=2
LogFilePages=4096
LogType=CIRCULAR
LogBufferPages=0
LogPath=/var/mqm/log/saturn!queue!manager/
```

Nota: Il valore di zero per **LogBufferPages** fornisce un valore di 512.

LogWriteIntegrity - utilizzando SingleWrite o TripleWrite

L'impostazione dell'opzione **LogWriteIntegrity** , nella stanza Log del file `qm.ini` , determina l'algoritmo utilizzato dal logger in IBM MQ per scrivere i record di log nel log di ripristino. L'impostazione predefinita è TripleWrite e questa impostazione è sicura in quasi tutti gli scenari possibili.

L'impostazione di **LogWriteIntegrity** ha alcun effetto, solo quando deve essere scritta una pagina di log parziale. Per un gestore code con una quantità ragionevole di attività simultanea, questo scenario si verifica raramente.

SingleWrite

SingleWrite seleziona un algoritmo che, in circostanze molto insolite, può essere eseguito meglio dell'impostazione predefinita TripleWrite . L'impostazione SingleWrite è sicura, solo se la piattaforma di archiviazione sottostante può assolutamente garantire in tutte le circostanze che le pagine 4KB scritte in modo sincrono nel log di recupero IBM MQ vengono scritte in modo atomico.

È necessario utilizzare l'impostazione SingleWrite , solo se il file - system o la periferica, che ospita il log di recupero IBM MQ , garantisce esplicitamente l'atomicità delle scritture 4KB . Vale a dire, quando una scrittura di una pagina 4KB ha esito negativo per qualsiasi motivo, gli unici due stati possibili devono essere l'immagine precedente o l'immagine successiva e non deve essere possibile alcuno stato intermedio. In tutti gli altri casi, utilizzare TripleWrite.

Su un sistema con una simultaneità sufficiente, il gestore code scrive solo pagine complete di dati di log e, se viene raggiunta un'elevata percentuale di pagine complete, non vi è alcuna differenza di prestazioni significativa tra SingleWrite e TripleWrite.

Su un sistema con poca simultaneità, può essere un vantaggio significativo sulle prestazioni per SingleWrite, tuttavia la soluzione preferita è di solito quella di aumentare la simultaneità, piuttosto che utilizzare SingleWrite.

Si noti che può essere difficile determinare in modo affidabile l'atomicità delle scritture 4KB e le modifiche al software o all'hardware sottostante potrebbero invalidare tale garanzia.

Se hai il dubbio che la tua infrastruttura di archiviazione offra le garanzie richieste ora e in qualsiasi momento in futuro in tutte le circostanze, dovresti utilizzare TripleWrite.

Windows Stanza LU62 del file qm.ini (soloWindows)

La stanza LU62 specifica parametri di configurazione del protocollo SNA LU 6.2 . Questi parametri sovrascrivono gli attributi predefiniti per i canali.

Utilizzare la sezione LU62 del file qm . ini per specificare i parametri di configurazione del protocollo SNA LU 6.2 . Sovrascrivono gli attributi predefiniti per i canali.

Linux

Windows

In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code IBM MQ Explorer LU6 . 2 .

TPName

Il nome TP da avviare sul sito remoto.

Library1= NomeDLL 1

Il nome della DLL APPC.

Il valore predefinito è WCPIC32.

Library2= DLLName2

Lo stesso di Library1, utilizzato se il codice è memorizzato in due librerie separate.

Il valore predefinito è WCPIC32.

CP4I

Stanza NativeHAInstance del file qm.ini

Per IBM MQ nei contenitori, la stanza NativeHAInstance specifica il modo in cui i tre nodi in una configurazione della HA nativa possono comunicare tra loro.

Nota: Queste informazioni si applicano solo agli ambienti contenitore. Consultare [Configurazione della HA nativa utilizzando l'operatore IBM MQ](#) o [Creazione del gruppo HA nativo se si creano i propri contenitori](#).

Si aggiungono tre stanze `NativeHAInstance` , una per ogni istanza del gestore code nel gruppo HA nativo (inclusa l'istanza locale). Aggiungere i seguenti attributi:

Nome

Specificare il nome istanza utilizzato quando è stata creata l'istanza del gestore code.

ReplicationAddress

Specificare il nome host, IPv4 decimale puntato o IPv6 l'indirizzo in formato esadecimale dell'istanza. È possibile specificare l'indirizzo come nome host, IPv4 decimale puntato o IPv6 indirizzo in formato esadecimale. L'indirizzo di replica deve essere risolvibile e instradabile da ogni istanza nel gruppo. Il numero di porta da utilizzare per la replica del log deve essere specificato tra parentesi, ad esempio:

```
ReplicationAddress=host1.example.com(4444)
```

Stanza di esempio

Il seguente esempio mostra la sezione di `NativeHAInstance` utilizzata nel file `qm.ini` per specificare i tre nodi di una configurazione HA nativa.

```
NativeHAInstance:  
  Name=node-1  
  ReplicationAddress=host1.example.com(4444)  
NativeHAInstance:  
  Name=node-2  
  ReplicationAddress=host2.example.com(4444)  
NativeHAInstance:  
  Name=node-3  
  ReplicationAddress=host3.example.com(4444)
```

Concetti correlati

“Stanza `NativeHALocalInstance` del file `qm.ini`” a [pagina 146](#)

Per IBM MQ nei contenitori, la stanza `NativeHALocalInstance` controlla il funzionamento di una configurazione della HA nativa.

CP4I

Stanza `NativeHALocalInstance` del file `qm.ini`

Per IBM MQ nei contenitori, la stanza `NativeHALocalInstance` controlla il funzionamento di una configurazione della HA nativa.

Nota: Queste informazioni si applicano solo agli ambienti contenitore. Consultare [Configurazione della HA nativa utilizzando l'operatore IBM MQ](#) o [Creazione del gruppo HA nativo](#) se si creano i propri contenitori.

La stanza `NativeHALocalInstance` viene aggiunta automaticamente al file `qm.ini` su ciascuno dei nodi quando crei una configurazione della HA nativa. È quindi possibile modificare il file `qm.ini` e modificare gli attributi nella stanza `NativeHALocalInstance` .

LocalName

Il nome della stanza `NativeHALocalInstance` , preso dal nome dell'istanza di replica del log specificato quando viene creato il gestore code della HA nativa.

Facoltativamente, è possibile aggiungere i seguenti attributi alla stanza `NativeHALocalInstance` :

KeyRepository

Il percorso completo e il nome file del contenitore chiavi che contiene il certificato digitale utilizzato per proteggere il traffico di replica del log. Se l'estensione file non è specificata, si presume che sia `.kdb`.

Se l'attributo della sezione `KeyRepository` viene omissso, i dati di replica del log vengono scambiati tra le istanze in testo semplice.

V9.4.0

KeyRepositoryPassword

Il repository delle chiavi è protetto con una password poiché contiene informazioni sensibili. Per poter accedere al contenuto del repository delle chiavi, IBM MQ deve essere in grado di richiamare la

password del repository delle chiavi. Se la password non è memorizzata in un file stash del repository delle chiavi, è possibile fornire la password nell'attributo `KeyRepositoryPassword`. Ad esempio:

```
KeyRepositoryPassword=passw0rd
```



Attenzione: Se si fornisce la password utilizzando questo attributo, codificare la password con il sistema di protezione password IBM MQ. Per ulteriori informazioni, consultare [“Codifica della password del repository delle chiavi”](#) a pagina 148.

V 9.4.0 **InitialKeyFile**

Specificare questo attributo se la password di repository delle chiavi specificata con l'attributo `KeyRepositoryPassword` è codificata con una chiave iniziale specifica. Il nome del file che contiene la chiave iniziale può essere specificato utilizzando il parametro **-sf** quando il comando **runmqicred** viene utilizzato per codificare la password del repository delle chiavi.

Impostare il valore di questo attributo sul nome del file che contiene la chiave iniziale utilizzata per codificare la password. Ad esempio, se un file denominato `mykey.key` contiene la chiave iniziale:

```
InitialKeyFile=/mykey.key
```

Per ulteriori informazioni, consultare [“Codifica della password del repository delle chiavi”](#) a pagina 148.

CertificateLabel

L'etichetta del certificato che indica il certificato digitale da utilizzare per la protezione del traffico di replica del log. Se `KeyRepository` viene fornito ma `CertificateLabel` viene omissso, viene utilizzato il valore predefinito `ibmwebspheremqueue_manager`.

CipherSpec

La `CipherSpec` da utilizzare per proteggere il traffico di replica del log. Se questo attributo della stanza viene fornito, è necessario fornire anche `KeyRepository`. Se `KeyRepository` viene fornito ma `CipherSpec` viene omissso, viene utilizzato il valore predefinito `ANY`.

LocalAddress

L'indirizzo dell'interfaccia di rete locale che accetta il traffico di replica del log. Se questo attributo della stanza viene fornito, identifica l'interfaccia di rete locale e / o la porta utilizzando il formato "[addr] [(port)]". L'indirizzo di rete può essere specificato come un nome host, in formato decimale puntato IPv4 o in formato esadecimale IPv6. Se questo attributo viene omissso, il gestore code tenta di collegarsi a tutte le interfacce di rete, utilizza la porta specificata in `ReplicationAddress` nella stanza `NativeHAInstances` che corrisponde al nome dell'istanza locale.

HeartbeatInterval

L'intervallo di heartbeat definisce la frequenza in millisecondi con cui un'istanza attiva di un gestore code HA nativo invia un heartbeat di rete. L'intervallo valido del valore dell'intervallo di heartbeat è compreso tra 500 (0.5 secondi) e 60000 (1 minuto), un valore esterno a questo intervallo causa l'errore di avvio del gestore code. Se questo attributo viene omissso, viene utilizzato un valore predefinito di 5000 (5 secondi). Ogni istanza deve utilizzare lo stesso intervallo di heartbeat.

HeartbeatTimeout

Il timeout heartbeat definisce il tempo di attesa di un'istanza di replica di un gestore code HA nativo prima di decidere che l'istanza attiva non risponde. L'intervallo valido del valore di timeout dell'intervallo di heartbeat è compreso tra 500 (0.5 secondi) e 120000 (2 minuti). Il valore del timeout di heartbeat deve essere maggiore o uguale all'intervallo di heartbeat.

Un valore non valido causa l'errore di avvio del gestore code. Se questo attributo viene omissso, una replica attende 2 x `HeartbeatInterval` prima di avviare il processo per selezionare una nuova istanza attiva. Ogni istanza deve utilizzare lo stesso timeout heartbeat.

RetryInterval

L'intervallo di nuovi tentativi definisce la frequenza in millisecondi con cui un gestore code HA nativo deve ritentare un link di replica non riuscito. L'intervallo valido per i tentativi è compreso tra 500 (0.5 secondi) e 120000 (2 minuti). Se questo attributo viene omissso, una replica attende 2 x `HeartbeatInterval` prima di ritentare un link di replica non riuscito.

SSLFipsRequired

Specifica se vengono utilizzati solo algoritmi certificati FIPS se viene utilizzata la crittografia nell'invio del traffico di replica del log. Impostare su Yes o No.

EncryptionPolicySuiteB

Specifica se il traffico di replica log utilizza la crittografia conforme a Suite - B e quale livello di complessità viene utilizzato. Impostare su uno dei seguenti valori:

NONE

La crittografia compatibile Suite - B non viene utilizzata. Questa è l'impostazione predefinita.

128_BIT,192_BIT

Imposta il livello di sicurezza su entrambi i livelli a 128 bit e 192 bit.

128_BIT

Imposta il livello di sicurezza a 128 bit.

192_BIT

Imposta il livello di sicurezza su 192 bit.

V 9.4.0

CompressionThreshold

Imposta una soglia di byte che, quando viene superata, attiva la compressione dei dati di log. I dati di log più grandi del valore di soglia sono compressi. Un valore pari a 0 (il valore predefinito) disattiva tutta la compressione, un valore pari a 1 comprime ogni accodamento di log. Il valore massimo è 268435456 (256 MB).

V 9.4.0

LZ4Acceleration

Un parametro di ottimizzazione che controlla il modo in cui l'algoritmo LZ4 ricerca le sequenze comprimibili nei dati di log. Ogni volta che il valore di accelerazione aumenta di 1, l'algoritmo non guarda con la stessa attenzione per una sequenza comprimibile, scambiando il rapporto di compressione per un piccolo guadagno di prestazione. Il valore minimo (e assunto predefinito) è 1, il massimo è 65537.

Codifica della password del repository delle chiavi

V 9.4.0

La password del repository delle chiavi può essere protetta utilizzando il sistema di protezione delle password IBM MQ o un file stash del repository delle chiavi. Per ulteriori informazioni su questi due metodi, consultare [Codifica delle password del repository delle chiavi](#).

Se la password del repository viene specificata utilizzando l'attributo `KeyRepositoryPassword` nella stanza `NativeHALocalInstance`, codificare la password utilizzando il sistema di protezione password IBM MQ. Utilizzare il comando **runmqicred** per codificare la parola d'ordine. Il comando restituisce la parola d'ordine codificata che può essere specificata nell'attributo `KeyRepositoryPassword`.

Utilizzare una chiave iniziale univoca per codificare la password in maniera sicura. Il nome del file che contiene la chiave iniziale può essere specificato utilizzando il parametro **-sf** nel comando **runmqicred**. Se non si fornisce una chiave univoca, viene utilizzata la chiave predefinita.

Se si codifica la password del repository delle chiavi con una chiave iniziale univoca, è necessario fornire anche la stessa chiave iniziale utilizzando l'attributo `InitialKeyFile` nella stanza `NativeHALocalInstance`.

Stanza di esempio

Il seguente esempio mostra la sezione `NativeHALocalInstance` utilizzata nel file `qm.ini` per specificare il nome locale di un nodo.

```
NativeHALocalInstance:  
  LocalName=node-1
```

Concetti correlati

[“Stanza NativeHAInstance del file qm.ini” a pagina 145](#)

Per IBM MQ nei contenitori, la stanza NativeHAInstance specifica il modo in cui i tre nodi in una configurazione della HA nativa possono comunicare tra loro.



Riferimenti correlati

[runmqicred \(proteggi password client IBM MQ\)](#)

Stanza NETBIOS del file qm.ini (soloWindows)

La stanza NETBIOS nel file `qm.ini` specifica i parametri di configurazione del protocollo NetBIOS. Questi parametri sovrascrivono gli attributi predefiniti per i canali.

Utilizzare la stanza NETBIOS nel file `qm.ini` per specificare i parametri di configurazione del protocollo NetBIOS. Sovrascrivono gli attributi predefiniti per i canali.

  In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code Netbios IBM MQ Explorer.

LocalName= nome

Il nome con cui questa macchina è nota sulla LAN.

AdapterNum = 0 (predefinito) | numero_adattatore

Il numero dell'adattatore LAN. Il valore predefinito è l'adattatore 0.

NumSess = 1 (valore predefinito) | numero_di_sessioni

Il numero di sessioni da assegnare. Il valore predefinito è 1.

NumCmds = 1 (valore predefinito) | numero_di_comandi

Il numero di comandi da assegnare. Il valore predefinito è 1.

NumNames = 1 (valore predefinito) | numero_di_nomi

Il numero di nomi da assegnare. Il valore predefinito è 1.

Library1= DLLName1

Il nome della DLL NetBIOS.

Il valore predefinito è NETAPI32.

Concetti correlati

[“Definizione del nome IBM MQ locale NetBIOS” a pagina 273](#)

Il nome NetBIOS locale utilizzato dai processi del canale IBM MQ può essere specificato in tre modi.

Stanza RestrictedMode del file qm.ini

La stanza RestrictedMode specifica il nome del gruppo che contiene i membri a cui è consentito eseguire le applicazioni MQI, aggiornare tutte le risorse IPCC e modificare il contenuto di alcune directory del gestore code. Questa stanza si applica solo ai sistemi AIX and Linux.

La stanza RestrictedMode è impostata dall'opzione **-g** sul comando **crtmqm**. Se non si utilizza l'opzione **-g**, la stanza non viene creata nel file `qm.ini`.

Alcune directory in cui le applicazioni IBM MQ creano file mentre sono connesse al gestore code all'interno della directory dei dati del gestore code. Per consentire alle applicazioni di creare i file in queste directory, viene loro concesso l'accesso in scrittura mondiale:

- `/var/mqm/sockets/QMgrName/@ipcc/ssem/hostname/`
- `/var/mqm/sockets/QMgrName/@app/ssem/hostname/`
- `/var/mqm/sockets/QMgrName/zsocketapp/hostname/`

dove `QMGRNAME` è il nome del gestore code e `hostname` è il nome host.

Su alcuni sistemi, non è possibile concedere a tutti gli utenti l'accesso in scrittura a tali directory. Ad esempio, gli utenti che non hanno bisogno di accedere al gestore code. La modalità limitata modifica le autorizzazioni delle directory che memorizzano i dati del gestore code. Le directory possono essere

accedute solo dai membri del gruppo di applicazioni specificato. Anche le autorizzazioni sulla memoria condivisa IPC System V utilizzate per comunicare con il gestore code vengono modificate nello stesso modo.

Il gruppo di applicazioni è il nome del gruppo con i membri che dispongono dell'autorizzazione per eseguire le seguenti operazioni:

- Esegui applicazioni MQI
- Aggiorna tutte le risorse IPCC
- Modificare il contenuto di alcune directory del gestore code


Per utilizzare la modalità limitata per un gestore code:

- Il creatore del gestore code deve essere nel gruppo mqm e nel gruppo di applicazioni.
- L'ID utente mqm deve essere nel gruppo di applicazioni.
- Tutti gli utenti che desiderano gestire il gestore code devono essere nel gruppo mqm e nel gruppo di applicazioni.
- Tutti gli utenti che desiderano eseguire applicazioni IBM MQ devono essere nel gruppo di applicazioni.

Qualsiasi chiamata MQCONN o MQCONNX emessa da un utente che non si trova nel gruppo di applicazioni ha esito negativo con codice motivo MQRC_Q_MGR_NOT_AVAILABLE.

Importante: Su molti sistemi operativi, affinché l'aggiunta di un utente a un gruppo venga riconosciuta, l'utente in questione deve scollegarsi e ricollegarsi.

La modalità con restrizioni funziona con il servizio di autorizzazione IBM MQ . Pertanto, è necessario concedere agli utenti anche l'autorità di connettersi a IBM MQ e accedere alle risorse richieste utilizzando il servizio di autorizzazione IBM MQ .

 Ulteriori informazioni sulla configurazione del servizio di autorizzazione IBM MQ sono disponibili in [Impostazione della sicurezza sui sistemi AIX, Linux, and Windows](#).

Utilizzare la modalità limitata IBM MQ solo quando il controllo fornito dal servizio di autorizzazione non fornisce un isolamento sufficiente delle risorse del gestore code.

Riferimenti correlati

[crtmqm](#) (crea gestore code)

Stanza di sicurezza del file qm.ini

La stanza Sicurezza specifica le opzioni per OAM (Object Authority Manager).

ClusterQueueAccessControl= RQMName | Xmitq

Impostare questo attributo per controllare il controllo accessi delle code cluster o delle code complete ospitate sui gestori code cluster.

RQMNAME

I profili controllati per il controllo accessi delle code ospitate in remoto sono le code denominate o i profili del gestore code.

XMITQ

I profili controllati per il controllo dell'accesso delle code ospitate in remoto vengono risolti in SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Xmitq è il valore predefinito.

GroupModel=GlobalGroups

Questo attributo determina se OAM controlla i gruppi globali quando determina l'appartenenza a un gruppo di utenti su Windows.

L'impostazione predefinita è di non controllare i gruppi globali.

GlobalGroups

OAM controlla i gruppi globali.

Con GlobalGroups impostato, i comandi di autorizzazione **setmqaut**, **dspmqaut** e **dmpmqaut** accettano i nomi dei gruppi globali; consultare il parametro **setmqaut -g**.



Nota: Se si imposta `ClusterQueueAccessControl=RQMName` e si dispone di una implementazione personalizzata del servizio di autorizzazione inferiore a `MQZAS_VERSION_6`, il gestore code non viene avviato. In questa istanza, impostare `ClusterQueueAccessControl=Xmitq` o aggiornare il servizio di autorizzazione personalizzato a `MQZAS_VERSION_6` o superiore.

Stanza di esempio

```
Security:
  ClusterQueueAccessControl=Xmitq
  GroupModel=GlobalGroups
```

Multi Stanza di servizio del file qm.ini

La stanza `Service` viene utilizzata per apportare modifiche ai servizi installabili. Questa stanza contiene il nome e il numero di punti di ingresso definiti per il servizio.

Nota:   La modifica dei servizi installabili e dei relativi componenti ha implicazioni significative. Per questo motivo, i servizi installabili sono di sola lettura in IBM MQ Explorer.

Per ogni componente all'interno di un servizio, è necessario anche specificare nome e percorso del modulo contenente il codice per tale componente. Utilizzare la stanza [ServiceComponent](#) per questo.

Le stanze **Service** e **ServiceComponent** possono essere presenti in qualsiasi ordine e le relative chiavi possono essere presenti anche in qualsiasi ordine. Per una di queste stanze, tutte le chiavi della stanza devono essere presenti. Se una chiave di stanza è duplicata, viene utilizzata l'ultima.



All'avvio, il gestore code elabora ciascuna voce del componente del servizio nel file di configurazione a turno. Quindi carica il modulo del componente specificato, richiamando il punto di ingresso del componente (che deve essere il punto di ingresso per l'inizializzazione del componente), passando ad esso un handle di configurazione.

Nome = AuthorizationService (predefinito) |NameService

Il nome del servizio richiesto.

AuthorizationService

Per IBM MQ, il componente **AuthorizationService** è noto come OAM (object authority manager). La stanza **Service** e la sua stanza **ServiceComponent** associata vengono aggiunte automaticamente quando il gestore code viene creato, ma possono essere sovrascritte dalla variabile di ambiente `MQSNOAUT`. Aggiungere manualmente altre **ServiceComponent** stanze.

  I seguenti esempi di stanze nel file `qm.ini` definiscono due componenti del servizio di autorizzazione su IBM MQ for AIX. `MQ_INSTALLATION_PATH` rappresenta la directory di livello superiore in cui è installato IBM MQ.

```
Service:
  Name=AuthorizationService
  EntryPoints=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=MQ_INSTALLATION_PATH/lib/amqzfu
  ComponentDataSize=0

ServiceComponent:
  Service=AuthorizationService
  Name=user.defined.authorization.service
  Module=/usr/bin/udas01
  ComponentDataSize=96
```

Linux AIX

La stanza ServiceComponent MQSeries.UNIX.auth.service definisce il componente del servizio di autorizzazione predefinito, OAM. Se si rimuove questa stanza e si riavvia il gestore code, l'OAM viene disabilitato e non viene eseguito alcun controllo di autorizzazione.

Windows

È anche possibile aggiungere l'attributo **SecurityPolicy** utilizzando i servizi IBM MQ. L'attributo **SecurityPolicy** si applica solo se il servizio specificato nella stanza Service è il servizio di autorizzazione, ossia l'OAM predefinito. L'attributo **SecurityPolicy** consente di specificare la politica di sicurezza per ciascun gestore code. I valori possibili sono:

Valore predefinito

Specificare Predefinito se si desidera che la politica di sicurezza predefinita abbia effetto. Se un identificativo di sicurezza Windows (SID NT) non viene passato all'OAM per un determinato ID utente, viene effettuato un tentativo di ottenere il SID appropriato ricercando i database di sicurezza pertinenti.

NTSIDsRequired

Richiede che un SID NT venga passato a OAM quando si eseguono i controlli di sicurezza.

Windows

La stanza ServiceComponent MQSeries.WindowsNT.auth.service definisce il componente del servizio di autorizzazione predefinito, OAM. Se si rimuove questa stanza e si riavvia il gestore code, l'OAM viene disabilitato e non viene eseguito alcun controllo di autorizzazione.

NameService

Per impostazione predefinita, non viene fornito alcun servizio nomi. Se si richiede un servizio dei nomi, è necessario aggiungere manualmente la sezione NameService.

Linux AIX

I seguenti esempi di stanze file AIX and Linux qm.ini per il servizio nomi specificano un componente del servizio nomi fornito dalla società ABC (fittizia).

```
# Stanza for name service
Service:
  Name=NameService
  EntryPoints=5

# Stanza for name service component, provided by ABC
ServiceComponent:
  Service=NameService
  Name=ABC.Name.Service
  Module=/usr/lib/abcname
  ComponentDataSize=1024
```

Nota: Windows

Sui sistemi Windows, le informazioni sulla sezione NameService vengono memorizzate nel registro.

EntryPoints= numero di voci

Il numero di punti di entrata definiti per il servizio.

Ciò include i punti di ingresso di inizializzazione e terminazione.

Windows

SecurityPolicy= Predefinito |NTSIDsRequired

Sui sistemi Windows, l'attributo **SecurityPolicy** si applica solo se il servizio specificato è il servizio di autorizzazione predefinito, ossia OAM. L'attributo **SecurityPolicy** consente di specificare la politica di sicurezza per ciascun gestore code.

I valori possibili sono:

Valore predefinito

Utilizzare la politica di sicurezza predefinita per rendere effettiva. Se un identificativo di sicurezza Windows (SID NT) non viene passato all'OAM per un determinato ID utente, viene effettuato un tentativo di ottenere il SID appropriato ricercando i database di sicurezza pertinenti.

NTSIDsRequired

Passare un SID NT a OAM quando si eseguono i controlli di sicurezza.

Per ulteriori informazioni, vedere [SID \(security identifier\)Windows](#).

Linux

AIX

SecurityPolicy=utente|gruppo|UserExternal|predefinito

Sui sistemi AIX and Linux , il valore specifica se il gestore code utilizza l'autorizzazione basata sull'utente o sul gruppo. I valori non sono sensibili al maiuscolo / minuscolo.

Il valore può essere uno dei seguenti:

gruppo

Il gestore code utilizza l'autorizzazione basata sul gruppo. L'autorizzazione ad accedere ad una risorsa è concessa ad un gruppo.

Un utente riceve l'aggregato di tutte le autorizzazioni concesse a ciascun gruppo a cui appartiene.

Gli ID utente e i gruppi devono essere definiti sul sistema operativo locale.

utente

Il gestore code utilizza l'autorizzazione basata sull'utente. L'autorizzazione per accedere a una risorsa può essere concessa a un gruppo o a un ID utente specifico.

Un utente riceve l'aggregato delle seguenti autorizzazioni:

- Autorizzazioni concesse all'utente specifico.
- Autorizzazioni concesse a ciascun gruppo a cui appartiene l'utente.

Gli ID utente e i gruppi devono essere definiti sul sistema operativo locale.

UserExternal

Il gestore code utilizza l'autorizzazione basata sull'utente. Tuttavia, le autorizzazioni possono essere concesse agli ID utente che non sono noti al sistema operativo locale.

L'autorizzazione per accedere a una risorsa può essere concessa a un gruppo o a un ID utente specifico.

Un utente riceve l'aggregato delle seguenti autorizzazioni:

- Autorizzazioni concesse all'utente specifico.
- Autorizzazioni concesse a ciascun gruppo a cui appartiene l'utente.

Se un utente non è noto al sistema operativo locale, viene considerato come appartenente solo al gruppo Nessuno. Per ulteriori informazioni sui gruppi, consultare [Principal e gruppi su AIX, Linux, and Windows](#). L'ID utente deve avere una lunghezza massima di 12 caratteri e deve essere conforme alle [regole per la denominazione degli oggetti IBM MQ](#).

È possibile modificare i gestori code esistenti per utilizzare questa opzione aggiuntiva senza perdere la configurazione corrente.

V 9.4.0

Questo è il valore predefinito se viene specificata la stanza AuthToken .

valore predefinito

Il gestore code utilizza l'autorizzazione basata sul gruppo. Il funzionamento è lo stesso dell'opzione group .

Questo è il valore predefinito se la stanza AuthToken non è specificata.

Riavviare il gestore code per rendere effettive le modifiche al valore dell'attributo.

Nota:

V 9.4.0

Linux

AIX

Da IBM MQ 9.4.0, se viene specificata la stanza AuthToken , il valore effettivo dell'attributo **SecurityPolicy** della stanza Service è impostato su UserExternal. L'autenticazione del token non è disponibile se **SecurityPolicy** è esplicitamente impostato su Gruppo nella stanza Service. Se **SecurityPolicy** è impostato su Gruppo, rimuovere l'attributo **SecurityPolicy** dalla sezione Servizio, quindi riavviare il gestore code. Per ulteriori informazioni, fare riferimento a [“Stanza AuthToken del file qm.ini” a pagina 121](#).

SharedBindingsUserId= tipo - utente

L'attributo **SharedBindingsUserId** si applica solo se il servizio specificato è il servizio di autorizzazione predefinito, ovvero l'OAM. L'attributo **SharedBindingsUserId** viene utilizzato solo in relazione ai bind condivisi. Questo valore consente di specificare se il campo *UserIdentifier* nella struttura *IdentityContext*, dalla funzione MQZ_AUTHENTICATE_USER, è l'ID utente effettivo o l'ID utente reale.

Per informazioni sulla funzione MQZ_AUTHENTICATE_USER, vedere [MQZ_AUTHENTICATE_USER - Authenticate user](#).

I valori possibili sono:

Valore predefinito

Il valore del campo *UserIdentifier* è impostato come ID utente reale.

Reale

Il valore del campo *UserIdentifier* è impostato come ID utente reale.

Effettivo

Il valore del campo *UserIdentifier* è impostato come ID utente effettivo.

FastpathBindingsUserId= tipo - utente

L'attributo **FastpathBindingsUserId** si applica solo se il servizio specificato è il servizio di autorizzazione predefinito, ovvero l'OAM. L'attributo **FastpathBindingsUserId** viene utilizzato solo in relazione ai collegamenti fastpath. Questo valore consente di specificare se il campo *UserIdentifier* nella struttura *IdentityContext*, dalla funzione MQZ_AUTHENTICATE_USER, è l'ID utente effettivo o l'ID utente reale.

Per informazioni sulla funzione MQZ_AUTHENTICATE_USER, vedere [MQZ_AUTHENTICATE_USER - Authenticate user](#).

I valori possibili sono:

Valore predefinito

Il valore del campo *UserIdentifier* è impostato come ID utente reale.

Reale

Il valore del campo *UserIdentifier* è impostato come ID utente reale.

Effettivo

Il valore del campo *UserIdentifier* è impostato come ID utente effettivo.

IsolatedBindingsUserId= tipo - utente

L'attributo **IsolatedBindingsUserId** si applica solo se il servizio specificato è il servizio di autorizzazione predefinito, ovvero l'OAM. L'attributo **IsolatedBindingsUserId** viene utilizzato solo in relazione ai bind isolati. Questo valore consente di specificare se il campo *UserIdentifier* nella struttura *IdentityContext*, dalla funzione MQZ_AUTHENTICATE_USER, è l'ID utente effettivo o l'ID utente reale.

Per informazioni sulla funzione MQZ_AUTHENTICATE_USER, vedere [MQZ_AUTHENTICATE_USER - Authenticate user](#).

I valori possibili sono:

Valore predefinito

Il valore del campo *UserIdentifier* è impostato come ID utente effettivo.

Reale

Il valore del campo *UserIdentifier* è impostato come ID utente reale.

Effettivo

Il valore del campo *UserIdentifier* è impostato come ID utente effettivo.

Per ulteriori informazioni sui servizi e i componenti installabili, consultare [Servizi e componenti installabili per AIX, Linux, and Windows](#).

Per ulteriori informazioni sui servizi di sicurezza in generale, consultare [Configurazione della sicurezza sui sistemi AIX and Linux](#).

Stanza di esempio

```
Service:  
  Name=AuthorizationService  
  EntryPoints=14
```

Concetti correlati

[Componenti e servizi installabili per AIX, Linux e Windows](#)

Riferimenti correlati

[Componenti e servizi installabili su IBM i](#)

[Informazioni di riferimento sui servizi installabili](#)

Multi

Stanza ServiceComponent del file qm.ini

La stanza ServiceComponent specifica le informazioni per il componente del servizio. È necessario specificare le informazioni sul componente del servizio quando viene aggiunto un nuovo servizio installabile. La stanza del servizio di autorizzazione è presente per impostazione predefinita e il componente associato, OAM, è attivo.

Le stanze **Service** e **ServiceComponent** possono essere presenti in qualsiasi ordine e le relative chiavi possono essere presenti anche in qualsiasi ordine. Per una di queste stanze, tutte le chiavi della stanza devono essere presenti. Se una chiave di stanza è duplicata, viene utilizzata l'ultima.

All'avvio, il gestore code elabora ciascuna voce del componente del servizio nel file di configurazione a turno. Quindi carica il modulo del componente specificato, richiamando il punto di ingresso del componente (che deve essere il punto di ingresso per l'inizializzazione del componente), passando ad esso un handle di configurazione.

Service = nome_servizio

Il nome del servizio richiesto. Deve corrispondere al valore specificato nell'attributo Name delle informazioni di configurazione del servizio.

Nome = nome_componente

Il nome descrittivo del componente servizio. Deve essere univoco e contenere solo caratteri validi per i nomi degli oggetti IBM MQ (ad esempio, nomi coda). Questo nome si verifica nei messaggi dell'operatore generati dal servizio. Si consiglia che questo nome inizi con un marchio aziendale o una stringa di distinzione simile.

Modulo = nome_modulo

Il nome del modulo che deve contenere il codice per questo componente. Questo deve essere un nome percorso completo.

ComponentDataDimensione = dimensione

La dimensione, in byte, dell'area dati del componente passata al componente su ciascuna chiamata. Specificare zero se non sono richiesti dati del componente.

Stanza di esempio

```
ServiceComponent:  
  Service=AuthorizationService  
  Name=MQSeries.UNIX.auth.service  
  Module=amqzfu  
  ComponentDataSize=0
```

Per ulteriori esempi che mostrano una stanza AuthorizationService e le stanze ServiceComponent associate e una stanza NameService e la stanza ServiceComponent associata, consultare [“Stanza di servizio del file qm.ini”](#) a pagina 151.

Concetti correlati

[Componenti e servizi installabili per AIX, Linux e Windows](#)

Riferimenti correlati

[“Stanza di servizio del file qm.ini” a pagina 151](#)

La stanza Service viene utilizzata per apportare modifiche ai servizi installabili. Questa stanza contiene il nome e il numero di punti di ingresso definiti per il servizio.



[Componenti e servizi installabili su IBM i](#)

[Informazioni di riferimento sui servizi installabili](#)

Stanza SPX del file qm.ini (soloWindows)

La stanza SPX specifica i parametri di configurazione del protocollo SPX. Questi parametri sovrascrivono gli attributi predefiniti per i canali.

Utilizzare la stanza SPX nel file `qm.ini` per specificare i parametri di configurazione del protocollo SPX.

  In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la pagina delle proprietà del gestore code IBM MQ Explorer SPX .

Socket = 5E86 (predefinito) | `socket_number`

Il numero di socket SPX in notazione esadecimale. Il valore predefinito è X'5E86'.

BoardNum = 0 (predefinito) | `numero_adattatore`

Il numero dell'adattatore LAN. Il valore predefinito è l'adattatore 0.

KeepAlive= NO | SÌ

Attivare o disattivare la funzione KeepAlive .

KeepAlive=YES fa sì che SPX controlli periodicamente che l'altra estremità della connessione sia ancora disponibile. In caso contrario, il canale viene chiuso.

Library1= `DLLName1`

Il nome della DLL SPX.

Il valore predefinito è WSOCK32.DLL.

Library2= `DLLName2`

Lo stesso di LibraryName1, utilizzato se il codice è memorizzato in due librerie separate.

Il valore predefinito è WSOCK32.DLL.

ListenerBacklog= numero

Sovrascrivere il numero predefinito di richieste in sospeso per il listener SPX.

Quando si riceve su SPX, viene impostato un numero massimo di richieste di connessione in sospeso. Questo può essere considerato un backlog di richieste in attesa sul socket SPX affinché il listener accetti la richiesta. I valori di backlog del listener predefiniti vengono mostrati in [Tabella 14 a pagina 156](#).

Piattaforma	Valore predefinito ListenerBacklog
Windows Server	100
Windows Workstation	5

Nota: Alcuni sistemi operativi supportano un valore maggiore del valore predefinito visualizzato. Utilizzare questa opzione per evitare di raggiungere il limite di connessione.

Al contrario, alcuni sistemi operativi potrebbero limitare la dimensione del backlog SPX, quindi il backlog SPX effettivo potrebbe essere più piccolo di quanto richiesto qui.

Se il backlog raggiunge i valori riportati in [Tabella 14 a pagina 156](#), la connessione SPX viene rifiutata e il canale non può essere avviato. Per i canali di messaggi, ciò fa sì che il canale entri in uno stato RETRY e ritenti la connessione in un secondo momento. Per connessioni client, il client riceve

un codice motivo MQRC_Q_MGR_NOT_AVAILABLE da MQCONN e deve ritentare la connessione in un secondo momento.

Multi Stanza SSL del file qm.ini

La stanza SSL viene utilizzata per configurare i canali TLS su un gestore code.

OCSP (Certificate Status Protocol Online)

Un certificato può contenere un'estensione AuthorityInfoAccess. Questa estensione specifica un server da contattare tramite OCSP (Online Certificate Status Protocol). Per consentire ai canali SSL o TLS sul gestore code di utilizzare le estensioni di accesso AuthorityInfo, assicurarsi che il server OCSP in essi indicato sia disponibile, configurato correttamente e accessibile sulla rete. Per ulteriori informazioni, consultare [Gestione dei certificati revocati](#).

CDP (CrIDistributionPoint)

Un certificato può contenere un'estensione punto CrIDistribution. Questa estensione contiene un URL che identifica sia il protocollo utilizzato per scaricare un CRL (Certificate Revocation List) sia il server da contattare.

Se vuoi consentire ai canali SSL o TLS sul tuo gestore code di utilizzare le estensioni del punto CrIDistribution, assicurati che il server CDP in essi indicato sia disponibile, configurato correttamente e accessibile sulla rete.

La stanza SSL

Utilizzare la stanza SSL nel file qm.ini per configurare il modo in cui i canali TLS sul gestore code tentano di utilizzare le funzioni riportate di seguito e il modo in cui reagiscono se si verificano problemi durante l'utilizzo.

In ciascuno dei seguenti casi, se il valore fornito non è uno dei valori validi elencati, viene utilizzato il valore predefinito. Non viene scritto alcun messaggio di errore che indica che è specificato un valore non valido.

OutboundSNI = CANALE | NOME HOST

Se **OutboundSNI** è impostato su CHANNEL, i client con capacità SNI impostano SNI sul nome canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS.

Se questo attributo è impostato su HOSTNAME, i client che supportano SNI impostano l'intestazione SNI sul nome host, causando la ricezione da parte delle richieste di connessione in uscita del certificato predefinito del gestore code remoto durante l'handshake TLS e quindi non è possibile utilizzare i certificati per canale.

Nota: Se **OutboundSNI**=HOSTNAME viene utilizzato per connettersi a un canale remoto con un'etichetta di certificato configurata, la connessione viene rifiutata con un messaggio MQRC_SSL_INITIALIZATION_ERROR e un messaggio [AMQ9673](#) viene stampato nei log di errore del gestore code remoto.

AllowOutboundSNI = YES (predefinito) | NO

Se abilitata, i client con capacità SNI imposteranno SNI sul nome del canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS. Se questo attributo è impostato su NO, i client con supporto SNI non imposteranno l'intestazione SNI causando la ricezione da parte delle richieste di connessione in uscita del certificato predefinito del gestore code remoto durante l'handshake TLS e, pertanto, i certificati per canale non possono essere utilizzati.



Attenzione: **Deprecated** Da IBM MQ 9.3.0 la proprietà **AllowOutboundSNI** è obsoleta ed è disponibile solo per scopi di compatibilità con le versioni precedenti.

AllowOutboundSNI impostato su YES fornisce la stessa funzione di **OutboundSNI** impostata su CHANNEL, mentre **AllowOutboundSNI** impostato su NO fornisce la stessa funzione di **OutboundSNI** impostata su HOSTNAME.

Se nella stanza SSL sono presenti sia gli attributi **AllowOutboundSNI** che **OutboundSNI**, l'impostazione di **OutboundSNI** ha la precedenza.

AllowedCipherSpecifiche =nome|elenco nomi| ALL

Specifica un elenco personalizzato di CipherSpecs ordinati e abilitati per l'utilizzo con canali IBM MQ su Multiplatforms.

- Un singolo nome CipherSpec .
- Un elenco separato da virgole di nomi IBM MQ CipherSpec da riabilitare.
- Il valore speciale di ALL, che rappresenta tutte le CipherSpecs (non consigliato).

Nota: Non si dovrebbe selezionare **ALL** CipherSpecs, poiché ciò abilita i protocolli SSL 3.0 e TLS 1.0 e un gran numero di algoritmi di crittografia deboli.

Per ulteriori informazioni, vedi [Fornitura di un elenco personalizzato di CipherSpecs ordinati e abilitati su IBM MQ for Multiplatforms nell'ordine CipherSpec nell'handshake TLS](#).

IBM I **ALW** **AllowTLSV13=Y | YES | T | TRUE (valore predefinito) | N | NO | F | FALSE**

Specifica se un gestore code è in grado di utilizzare TLS 1.3 CipherSpecs.

- Y (predefinito), YES (predefinito), T (predefinito) o TRUE (predefinito): abilita TLS 1.3 che consente al gestore code di utilizzare TLS 1.3 CipherSpecs.
- N, NO, F o FALSE: disabilita TLS 1.3, il che significa che il gestore code non è in grado di utilizzare i CipherSpecs TLS 1.3 .

Per ulteriori informazioni, consultare [Abilitazione di CipherSpecs](#).

CDPCheckExtensions= SÌ |NO (predefinito)

Specifica se i canali TLS su questo gestore code tentano di controllare i server CDP denominati nelle estensioni certificato del punto CrlDistribution.

- YES: i canali TLS tentano di controllare i server CDP per determinare se un certificato digitale viene revocato.
- NO (predefinito): i canali TLS non tentano di verificare i server CDP. Questo è il valore predefinito.

V 9.4.0 **HTTPSKeyStore= stringa**

La stringa fornisce il percorso di un repository delle chiavi pkcs12 che può essere utilizzato dal gestore code come truststore durante la creazione di connessioni https in uscita, ad esempio per un endpoint JWKS.

Questo file deve essere codificato e deve essere accompagnato da un file 'stash' con lo stesso nome, ossia un file con estensione .sth , utilizzato quando il gestore code deve accedere a questo file. Per impostazione predefinita, se questo attributo non viene specificato, il gestore code cerca un file denominato mqdefcer.p12 nella sottodirectory ssl dei dati del gestore code.

Se questo attributo viene aggiornato, è necessario riavviare il gestore code per iniziare a utilizzare il nuovo file del repository delle chiavi specificato.

Consultare [Creazione di un repository di chiavi da utilizzare come truststore TLS](#) per una guida sulla creazione di un truststore basato sull'ambiente del sistema operativo.

ALW **MinimumRSAKeySize=int**

Specifica la dimensione chiave minima che i certificati RSA devono avere per essere accettati durante un handshake TLS. Consente qualsiasi valore uguale o superiore a 0. Se non specificato, il valore predefinito è 1.

OCSPAuthentication=REQUIRED (predefinito) | WARN | OPTIONAL

Specifica l'azione da intraprendere quando uno stato di revoca non può essere determinato da un server OCSP.

Se il controllo OCSP è abilitato, un programma del canale TLS tenta di contattare un server OCSP.

Se il programma del canale non è in grado di contattare alcun server OCSP, o se nessun server può fornire lo stato di revoca del certificato, viene utilizzato il valore del parametro OCSPAuthentication.

- **REQUIRED** (valore predefinito): l'errore di determinazione dello stato di revoca causa la chiusura della connessione con errore. Questo è il valore predefinito.
- **WARN**: se non si determina lo stato di revoca, viene scritto un messaggio di avviso nel log degli errori del gestore code, ma la connessione può continuare.
- **FACOLTATIVO**: l'errore nel determinare lo stato di revoca consente alla connessione di procedere in modalità non presidiata. Non vengono forniti avvisi o errori.

OCSPCheckExtensions = YES (valore predefinito) | NO

Specifica se i canali TLS su questo gestore code tentano di controllare i server OCSP denominati nelle estensioni certificato di accesso AuthorityInfo.

- **YES** (valore predefinito): i canali TLS provano a controllare i server OCSP per determinare se un certificato digitale è revocato. Questo è il valore predefinito.
- **NO**: i canali TLS non tentano di controllare i server OCSP.

ALW **OCSPTimeout= numero**

Il numero di secondi di attesa per un responder OCSP durante l'esecuzione di una verifica di revoca.

Da IBM MQ 9.3.0, se è impostato il valore 0, viene utilizzato il timeout predefinito di 30 secondi.

Se non viene impostato alcun valore, viene utilizzato il valore predefinito IBM MQ di 30 secondi.

SSLHTTPProxyName= stringa

La stringa è il nome host o l'indirizzo di rete del server proxy HTTP che deve essere utilizzato da IBM Global Security Kit (GSKit) per i controlli OCSP. Questo indirizzo può essere seguito da un numero di porta facoltativo, racchiuso tra parentesi. Se non si specifica alcun numero, viene utilizzata la porta HTTP predefinita (80).

AIX Per client a 32 bit su AIX, l'indirizzo di rete può essere solo IPv4 .

Su altre piattaforme, l'indirizzo di rete può essere un indirizzo IPv4 o IPv6 .

Questo attributo potrebbe essere necessario se, ad esempio, un firewall impedisce l'accesso all'URL del responder OCSP.

ALW **PeerCertChainValidation=stringa**

La stringa può essere uno dei due valori:

- **Usepeerchain [Valore predefinito]**: la catena di certificati fornita dal peer può essere utilizzata per colmare eventuali lacune della catena di attendibilità durante la convalida dei certificati. Con l'eccezione del certificato root.
- **Truststoreonly [Non consigliato]**: verranno utilizzati solo i certificati nel truststore per convalidare il certificato del peer.

ALW **SSLHTTPConnectTimeout= numero|0**

Il numero di secondi di attesa per stabilire correttamente una connessione di rete a un server HTTP durante l'esecuzione di un controllo di revoca.

Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ 0 (off).

Stanza di esempio

```
SSL:
  OutboundSNI=CHANNEL
  AllowedCipherSpecs=TLS13 CipherSpec list
  AllowTLSV13=Y
  CDPCheckExtensions=NO
  MinimumRSAKeySize=1
  OCSPAuthentication=REQUIRED
  OCSPCheckExtensions=YES
  OCSPTimeout=30
  PeerCertChainValidation=Usepeerchain
  SSLHTTPConnectTimeout=0
```

Note:

- Il valore predefinito per **OutboundSNI** è **Channel1**.
- L'elenco **TLS13 CipherSpec** è un elenco di specifiche CipherSpecs non le cifrature alias. Se si richiedono solo cifrature TLS1.3 , è necessario elencarle. Ad esempio:

```
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_256_GCM_SHA384
TLS_AES_128_GCM_SHA256
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256
```

- Il valore predefinito per **AllowTLSV13** è Y a meno che non siano state abilitate le cifrature deboli, nel qual caso viene disattivato (a meno che non venga esplicitamente attivato).
- I valori per **CDPCheckExtensions** possono essere solo Sì o No.
- I valori per **PeerCertChainValidation** possono essere solo Usepeerchain o Truststoreonly.

Multi

Stanza del pool secondario del file qm.ini

Questa stanza è creata da IBM MQ. Non modificarlo.

La stanza del pool secondario e l'attributo **ShortSubpoolName** all'interno di tale stanza vengono scritti automaticamente da IBM MQ quando si crea un gestore code. IBM MQ sceglie un valore per **ShortSubpoolName**. Non modificare questo valore.

Il nome corrisponde a una directory e a un collegamento simbolico creati all'interno della directory `/var/mqm/sockets` , che IBM MQ utilizza per le comunicazioni interne tra i suoi processi in esecuzione.

Multi

Stanza TCP del file qm.ini

La stanza TCP specifica i parametri di configurazione TCP/IP (Transmission Control Protocol/Internet Protocol). Questi parametri sovrascrivono gli attributi predefiniti per i canali.

Utilizzare la sezione TCP nel file `qm.ini` per specificare i parametri di configurazione TCP/IP.

Linux

Windows

In alternativa, in Linux (x86 e x86-64) e in Windows, utilizzare la pagina delle proprietà del gestore code TCP IBM MQ Explorer SPX .

Porta = 1414 (predefinito) | *numero_porta*

Il numero di porta predefinito, in notazione decimale, per le sessioni TCP/IP. Il numero di porta *ben noto* per IBM MQ è 1414.

Windows

Library1= *DLLName1* (soloWindows)

Il nome della DLL socket TCP/IP.

Il valore predefinito è WSOCK32.

Multi SecureCommsOnly = NO | N | FALSE | F (valore predefinito) | TRUE | T | YES |

Y

Specificare se la comunicazione in testo semplice è consentita o meno.

SecureCommsOnly=NO | N | FALSE | F

È consentita la comunicazione in testo semplice e viene emesso un messaggio di avviso all'avvio del gestore code.

SecureCommsOnly=YES | Y | TRUE | T

La comunicazione in testo semplice non è consentita e viene emesso un messaggio informativo all'avvio del gestore code.

KeepAlive = NO (valore predefinito) | SÌ

Attivare o disattivare la funzione KeepAlive . KeepAlive=YES fa sì che TCP/IP controlli periodicamente che l'altra estremità della connessione sia ancora disponibile. In caso contrario, il canale viene chiuso.

ListenerBacklog= numero

Sovrascrivere il numero predefinito di richieste in sospeso per il listener TCP/IP.

Quando si riceve su TCP/IP, viene impostato un numero massimo di richieste di connessione in sospeso. Questo può essere considerato come un backlog di richieste in attesa sulla porta TCP/IP affinché il listener accetti la richiesta. I valori di backlog del listener predefiniti vengono mostrati in [Tabella 15 a pagina 161](#).

Piattaforma	Valore predefinito ListenerBacklog
Windows ServerWindows	100
Linux Linux	100
AIX AIX V5.3 o successive	100

Nota: Alcuni sistemi operativi supportano un valore maggiore del valore predefinito visualizzato. Utilizzare questa opzione per evitare di raggiungere il limite di connessione.

Al contrario, alcuni sistemi operativi potrebbero limitare la dimensione del backlog TCP, quindi il backlog TCP effettivo potrebbe essere più piccolo di quanto richiesto qui.

Se il backlog raggiunge i valori mostrati in [Tabella 15 a pagina 161](#), la connessione TCP/IP viene rifiutata e non è possibile avviare il canale. Per i canali di messaggi, ciò fa sì che il canale entri in uno stato RETRY e ritenti la connessione in un secondo momento. Per le connessioni client, il client riceve un codice motivo MQRC_Q_MGR_NOT_AVAILABLE da MQCONN e ritenta la connessione successivamente.

Il seguente gruppo di proprietà può essere utilizzato per controllare la dimensione dei buffer utilizzati da TCP/IP. I valori vengono passati direttamente al livello TCP/IP del sistema operativo. Prestare molta attenzione quando si utilizzano queste proprietà. Se i valori non sono impostati correttamente, possono influire negativamente sulle prestazioni TCP/IP. Per ulteriori informazioni su come ciò influisce sulle prestazioni, fare riferimento alla documentazione TCP/IP per il proprio ambiente. Un valore zero indica che il sistema operativo gestirà le dimensioni del buffer, in contrapposizione alle dimensioni del buffer fissate da IBM MQ.

Connect_Timeout = 0 (valore predefinito) | numero

Il numero di secondi prima del timeout di un tentativo di connessione del socket. Il valore predefinito zero specifica che non esiste alcun timeout di connessione.

I processi del canale IBM MQ si collegano su socket non bloccanti. Pertanto, se l'altra estremità del socket non è pronta, connect () restituisce immediatamente EINPROGRESS o EWOULDBLOCK. In seguito, la connessione verrà tentata di nuovo, fino a un totale di 20 tentativi, quando viene notificato un errore di comunicazione.

Se `Connect_Timeout` è impostato su un valore diverso da zero, IBM MQ attende il periodo stabilito per la chiamata `select ()` affinché il socket sia pronto. Ciò aumenta le possibilità di riuscita di una chiamata `connect ()` successiva. Questa opzione potrebbe essere utile in situazioni in cui le connessioni richiedono un periodo di attesa, a causa di un carico elevato sulla rete.

SndBufferSize = numero |0 (valore predefinito)

La dimensione in byte del buffer di invio TCP/IP utilizzato dalla fine di invio dei canali. Questo valore di stanza può essere sovrascritto da una stanza più specifica per il tipo di canale, ad esempio `RcvSndBufferSize`. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Multi I nuovi gestori code vengono creati automaticamente con un'impostazione predefinita di 0 (consultare [“Stanza di esempio”](#) a pagina 162).

RcvBufferSize = numero |0 (predefinito)

La dimensione in byte del buffer di ricezione TCP/IP utilizzato dall'estremità di ricezione dei canali. Questo valore di stanza può essere sovrascritto da una stanza più specifica per il tipo di canale, ad esempio `RcvRcvBufferSize`. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Multi I nuovi gestori code vengono creati automaticamente con un'impostazione predefinita di 0 (consultare [“Stanza di esempio”](#) a pagina 162).

RcvSndBufferSize = numero|0 (valore predefinito)

La dimensione in byte del buffer di invio TCP/IP utilizzato dall'estremità mittente di un canale ricevente. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Multi I nuovi gestori code vengono creati automaticamente con un'impostazione predefinita di 0 (consultare [“Stanza di esempio”](#) a pagina 162).

RcvRcvBufferSize = numero |0 (valore predefinito)

La dimensione in byte del buffer di ricezione TCP/IP utilizzato dall'estremità di ricezione di un canale ricevente. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Multi I nuovi gestori code vengono creati automaticamente con un'impostazione predefinita di 0 (consultare [“Stanza di esempio”](#) a pagina 162).

SvrSndBufferSize = numero |0 (valore predefinito)

La dimensione in byte del buffer di invio TCP/IP utilizzato dall'estremità del server di un canale di connessione server di connessione client. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Multi I nuovi gestori code vengono creati automaticamente con un'impostazione predefinita di 0 (consultare [“Stanza di esempio”](#) a pagina 162).

SvrRcvBufferSize = numero |0 (valore predefinito)

La dimensione in byte del buffer di ricezione TCP/IP utilizzato dall'estremità server di un canale di connessione server di connessione client. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Multi I nuovi gestori code vengono creati automaticamente con l'impostazione predefinita 0 (consultare [“Stanza di esempio”](#) a pagina 162).

Stanza di esempio

TCP :

```
SndBuffSize=0
RcvBuffSize=0
RcvSndBuffSize=0
RcvRcvBuffSize=0
ClntSndBuffSize=0
ClntRcvBuffSize=0
SvrSndBuffSize=0
SvrRcvBuffSize=0
```

Nota: **Multi** Per i nuovi gestori code su Multiplatforms, le dimensioni buffer di invio e ricezione TCP predefinite nella stanza TCP di `qm.ini` file sono impostate per essere gestite dal sistema operativo. Come mostrato nell'esempio precedente, i nuovi gestori code vengono creati automaticamente con l'impostazione predefinita 0 per i buffer di invio e ricezione. Ciò si applica solo ai gestori code nuovi. Le impostazioni del buffer di invio e ricezione TCP per i gestori code migrati dalle precedenti versioni di IBM MQ vengono conservate.

Se le proprietà della dimensione del buffer TCP vengono rimosse dal file `qm.ini`, il buffer predefinito è impostato su 32K. È necessario prestare attenzione quando si utilizza questo valore predefinito poiché 32K potrebbe non essere un buffer appropriato per tutti gli scenari di messaggistica.

Se le proprietà del buffer di invio e ricezione TCP sono impostate su zero, vengono utilizzati i valori predefiniti del sistema operativo. Il metodo per la scelta di questi valori predefiniti varia a seconda del sistema operativo, ma in genere può essere trovato nelle pagine del manuale del sistema operativo "tcp" o `get/setsockopt ()`.

Multi Stanza TuningParameters del file qm.ini

La stanza TuningParameters specifica le opzioni per l'ottimizzazione del gestore code.

SuppressDspAuthFail= YES |NO (valore predefinito)

Quando è impostato su YES, il gestore code sopprime la creazione di eventi di autorizzazione e la scrittura di messaggi di errore `AMQ8077` nel log degli errori quando un controllo di autorizzazione ha esito negativo, se la connessione non dispone dell'autorizzazione + dsp per un oggetto.

ImplSyncOpenOutput=valore

ImplSyncOpenOutput è il numero minimo di applicazioni che hanno la coda aperta per l'inserimento, prima che un punto di sincronizzazione implicito possa essere abilitato per un inserimento persistente, al di fuori del punto di sincronizzazione. Il valore predefinito di **ImplSyncOpenOutput** è 2.

Ciò ha l'effetto che se esiste solo un'applicazione che ha quella coda aperta per un'operazione di inserimento, **ImplSyncOpenOutput** viene disattivato.

Specificare `ImplSyncOpenOutput=1` significa che viene sempre considerato un punto di sincronizzazione implicito. È possibile impostare qualsiasi numero intero positivo. Se non si desidera mai aggiungere un punto di sincronizzazione implicito, impostare `ImplSyncOpenOutput=OFF`.

UniformClusterNome =nome del cluster

Il nome del cluster IBM MQ che si sta utilizzando come cluster uniforme.

OAMLdapConnectTimeout=ora|0 (predefinito)

Il tempo massimo, in secondi, che il client LDAP attenderà per stabilire una connessione TCP al server. Se si stanno fornendo più server LDAP tramite un elenco nomi di connessione, il timeout si applica a ogni singolo tentativo di connessione, e quindi si tenta una connessione alla voce successiva nell'elenco nomi se questo timeout viene raggiunto.

`time` ha un valore massimo di 3600 secondi e un valore di 0, che è il valore minimo e il valore predefinito, significa che l'attesa è illimitata.

OAMLdapQueryTimeLimit=ora|0 (predefinito)

Il tempo massimo, in secondi, che il client LDAP attende per ricevere una risposta a una richiesta LDAP dal server, una volta stabilita una connessione e inviata una richiesta LDAP.

`time` ha un valore massimo di 3600 secondi e un valore di 0, che è il valore minimo e il valore predefinito, significa che l'attesa è illimitata.

OAMLdapResponseWarningTime=soglia

Se una connessione a un server LDAP ha impiegato più tempo del numero di soglia di secondi specificato dal parametro **OAMLdapResponseWarningTime** , un messaggio `AMQ5544W` verrà scritto nel log degli errori. La soglia predefinita è 10 secondi.

ExpiryInterval

Indica la frequenza con cui il gestore code esegue la scansione delle code alla ricerca di messaggi scaduti che non sono stati già ripuliti da altre attività della coda. Si tratta di un intervallo di tempo in secondi.

Per impostazione predefinita, lo scanner di scadenza viene eseguito approssimativamente ogni 5 minuti sulle build IBM MQ di produzione.



Avvertenza: La modifica del valore **ExpiryInterval** non è generalmente richiesta ed è necessario modificare questo valore solo sotto la guida del supporto IBM .

LivenessHeartBeatLen

Configura la frequenza con cui i controlli del gestore code che scrivono nel log vengono eseguiti ad una velocità ragionevole. Il valore massimo per **LivenessHeartBeatLen** è 600 secondi (10 minuti) e il valore minimo è 0, che ha l'effetto di disabilitare del tutto il controllo.



Avvertenza: Nella maggior parte dei casi, non è necessario modificare la frequenza di tali controlli. Non apportare alcuna modifica a meno che non sia consigliato dal supporto IBM .

ECHeartBeatLen

Configura la frequenza dei controlli di integrità generali del gestore code. Il valore minimo per **ECHeartBeatLen** è 10000 millisecondi (10 secondi) e il valore massimo è 60000 millisecondi (60 secondi).



Avvertenza: Nella maggior parte dei casi, non è necessario modificare la frequenza di tali controlli. Non apportare alcuna modifica a meno che non sia consigliato dal supporto IBM .

FileLockHeartBeatLen

Modifica il valore predefinito per i controlli del blocco file per un gestore code a più istanze che il controller di esecuzione esegue periodicamente per assicurarsi che mantenga ancora il blocco esclusivo sul file a più istanze primario. Per impostazione predefinita, questi controlli di blocco file vengono eseguiti ogni 20 secondi. Il valore minimo per **FileLockHeartBeatLen** è 10 secondi e il valore massimo è 600 secondi (10 minuti).



Avvertenza: Nella maggior parte dei casi, non è necessario modificare la frequenza di tali controlli. Non apportare alcuna modifica a meno che non sia consigliato dal supporto IBM .

Stanza di esempio

```
TuningParameters:
  SuppressDspAuthFail=NO
  ImplSyncOpenOutput=2
  OAMLdapConnectTimeout=60
  OAMLdapQueryTimeLimit=60
  OAMLdapResponseWarningTime=10
  ExpiryInterval=300
```

Concetti correlati

Punto di sincronizzazione implicito

Multi Stanza delle variabili del file qm.ini

La stanza Variabili specifica le variabili di configurazione da utilizzare con i cluster uniformi automatici.

È possibile utilizzare gli attributi elencati nella stanza Variables durante la configurazione cluster automatica di CONNAME e i campi MQSC del nome canale di un canale ricevente del cluster. Le variabili di configurazione non possono essere utilizzate in nessun altro elemento di uno script MQSC.

attributo=valore

Specifica un nome e il valore associato da utilizzare come inserimento durante le definizioni MQSC.

Le coppie *attribute=value* possono essere fornite utilizzando l'opzione della riga comandi **-iv** nel comando **crtmqm** quando si crea un gestore code.

Stanza di esempio

```
Variables:  
CONNAME=127.0.0.1(1414)
```

Concetti correlati

[“Bilanciamento automatico dell'applicazione” a pagina 428](#)

Il bilanciamento automatico delle applicazioni migliora notevolmente la distribuzione e la disponibilità delle applicazioni, consentendo a un cluster uniforme IBM MQ di gestire da vicino la distribuzione delle applicazioni nel cluster, piuttosto che affidarsi alla randomizzazione o al blocco manuale delle applicazioni a specifici gestori code.

Attività correlate

[“Creazione di un nuovo cluster uniforme” a pagina 442](#)

Come creare un nuovo cluster uniforme.

Riferimenti correlati

[“Utilizzo della configurazione automatica del cluster” a pagina 446](#)

Configurare IBM MQ per abilitare la configurazione automatica modificando le informazioni di configurazione `qm.ini`.

Multi

Stanza XAResourceManager del file qm.ini

La stanza XAResourceManager per specificare le informazioni sui gestori risorse coinvolti nelle unità di lavoro globali coordinate dal gestore code.

Utilizzare la stanza XAResourceManager nel file `qm.ini` per specificare le informazioni sui gestori risorse coinvolti nelle unità di lavoro globali coordinate dal gestore code.

Linux

Windows

In alternativa, su Linux (x86 e x86-64) e Windows, utilizzare la IBM MQ Explorer pagina delle propriet ... del gestore code XA.

Aggiungere manualmente le informazioni di configurazione del gestore risorse XA per ogni istanza di un gestore risorse che partecipa alle unità di lavoro globali; non vengono forniti valori predefiniti.

Consultare [Coordinamento database](#) per ulteriori informazioni sugli attributi del gestore risorse.

Nome = nome (obbligatorio)

Questo attributo identifica l'istanza del gestore risorse.

Il valore Name può contenere un massimo di 31 caratteri. È possibile utilizzare il nome del gestore risorse come definito nella relativa struttura XA - switch. Tuttavia, se si utilizza più di un'istanza dello stesso gestore risorse, è necessario creare un nome univoco per ciascuna istanza. È possibile garantire l'univocità includendo il nome del database nella stringa Name, ad esempio.

IBM MQ utilizza il valore Name nei messaggi e nell'output del comando `dspmqrtn`.

Non modificare il nome di un'istanza del gestore risorse o eliminare la relativa voce dalle informazioni di configurazione, una volta avviato il gestore code associato e attivato il nome del gestore risorse.

SwitchFile= nome (obbligatorio)

Il nome completo del file di caricamento contenente la struttura dello switch XA del gestore risorse.

Se si sta utilizzando un gestore code a 64 bit con applicazioni a 32 bit, il valore name deve contenere solo il nome di base del file di caricamento che contiene la struttura dello switch XA del gestore risorse.

Il file a 32 bit verrà caricato nell'applicazione dal percorso specificato da `ExitsDefaultPath`.

Il file a 64 bit verrà caricato nel gestore code dal percorso specificato da `ExitsDefaultPath64`.

XAOpenString= stringa (facoltativo)

La stringa di dati da passare al punto di ingresso `xa_open` del gestore risorse. Il contenuto della stringa dipende dal gestore risorse stesso. Ad esempio, la stringa potrebbe identificare il database a cui deve accedere questa istanza del gestore risorse. Per ulteriori informazioni sulla definizione di questo attributo, consultare:

- [Aggiunta delle informazioni di configurazione del gestore risorse per Db2](#)
- [Aggiunta di informazioni di configurazione del gestore risorse per Oracle](#)
- [Aggiunta di informazioni di configurazione del gestore risorse per Sybase](#)
- [Aggiunta delle informazioni di configurazione del gestore risorse per Informix](#)

e consultare la documentazione del gestore risorse per la stringa appropriata.

XACloseString= stringa (facoltativo)

La stringa di dati da trasmettere al punto di ingresso `xa_close` del gestore risorse. Il contenuto della stringa dipende dal gestore risorse stesso. Per ulteriori informazioni sulla definizione di questo attributo, consultare:

- [Aggiunta delle informazioni di configurazione del gestore risorse per Db2](#)
- [Aggiunta di informazioni di configurazione del gestore risorse per Oracle](#)
- [Aggiunta di informazioni di configurazione del gestore risorse per Sybase](#)
- [Aggiunta delle informazioni di configurazione del gestore risorse per Informix](#)

e consultare la documentazione del database per la stringa appropriata.

Controllo ThreadOf= THREAD | PROCESS

Windows Questo attributo è obbligatorio per Windows. Il gestore code utilizza questo valore per la serializzazione quando deve richiamare il gestore risorse da uno dei propri processi a più thread.

THREAD

Il gestore risorse è completamente *thread aware*. In un processo IBM MQ a più thread, è possibile effettuare chiamate di funzioni XA al gestore risorse esterno da più thread contemporaneamente.

PROCESS

Il gestore risorse non è *thread safe*. In un processo IBM MQ a più thread, è possibile effettuare una sola chiamata di funzione XA alla volta al gestore risorse.

La voce **ThreadOfControl** non si applica alle chiamate alla funzione XA emesse dal gestore code in un processo dell'applicazione a più thread. In generale, un'applicazione che ha unità di lavoro simultanee su thread differenti richiede che questa modalità di operazione sia supportata da ciascuno dei gestori risorse.

Stanza di esempio

```
XAResourceManager:  
Name=DB2 Resource Manager Bank  
SwitchFile=/usr/bin/db2swit  
XAOpenString=MQBankDB  
XACloseString=  
ThreadOfControl=THREAD
```

Nota: Il numero massimo di stanze XAResourceManager è limitato a 255. Tuttavia, è necessario utilizzare solo un numero ridotto di stanze per evitare la riduzione delle prestazioni della transazione.

IBM i File qm.ini di esempio per IBM i

Un esempio che mostra il modo in cui i gruppi di attributi possono essere organizzati in un file di configurazione del gestore code per IBM i.

```
#####  
#* Module Name: qm.ini *#  
#* Type : IBM MQ queue manager configuration file *#  
#* Function : Define the configuration of a single queue manager *#  
#* *#  
#####  
#* Notes : *#  
#* 1) This file defines the configuration of the queue manager *#  
#* *#  
#####  
Log:  
LogPath=QMSATURN.Q  
LogReceiverSize=65536  
  
CHANNELS:  
MaxChannels = 20 ; Maximum number of channels allowed.  
 ; Default is 100.  
MaxActiveChannels = 10 ; Maximum number of channels allowed to be  
 ; active at any time. The default is the  
 ; value of MaxChannels.  
  
TCP: ; TCP/IP entries.  
KeepAlive = Yes ; Switch KeepAlive on.  
SvrSndBuffSize=20000 ; Size in bytes of the TCP/IP send buffer for each  
 ; channel instance. Default is 32768.  
SvrRcvBuffSize=20000 ; Size in bytes of the TCP/IP receive buffer for each  
 ; channel instance. Default is 32768.  
Connect_Timeout=10000 ; Number of seconds before an attempt to connect the  
 ; channel instance times out. Default is zero (no timeout).  
  
QMErrorLog:  
ErrorLogSize = 262144  
ExcludeMessage = 7234  
SuppressMessage = 9001,9002,9202  
SuppressInterval = 30  
  
TuningParameters:  
ImplSyncOpenOutput=2
```

ALW File di configurazione dell'installazione, mqinst.ini

Sui sistemi AIX and Linux , il file di configurazione dell'installazione, mqinst .ini, contiene informazioni su tutte le installazioni di IBM MQ . Su sistemi Windows , le informazioni di configurazione dell'installazione si trovano nel registro.

Ubicazione del file mqinst .ini

Linux AIX

Il file mqinst .ini si trova nella directory /etc/opt/mqm sui sistemi AIX and Linux . Contiene informazioni su quale installazione, se presente, è l'installazione primaria e le seguenti informazioni per ciascuna installazione:

- Il nome dell'installazione
- La descrizione dell'installazione
- L'identificativo di installazione
- Il percorso di installazione

Importante: Il file mqinst .ini non deve essere modificato o a cui si fa riferimento direttamente poiché il formato non è fisso e potrebbe essere modificato.

L'identificativo di installazione, solo per uso interno, è impostato automaticamente e non deve essere modificato.

Invece di modificare direttamente il file `mqinst.ini`, è necessario utilizzare i seguenti comandi per creare, eliminare, interrogare e modificare i valori nel file:

`crtmqinst` per creare voci.

`dltmqinst` per eliminare le voci.

`dspmqinst` per visualizzare le voci.

`setmqinst` per impostare le voci.

Informazioni sulla configurazione dell'installazione su Windows

Windows

Non esiste alcun file `mqinst.ini` su Windows. Le informazioni di configurazione dell'installazione si trovano nel registro e si trovano nella seguente chiave:

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

Importante: Questa chiave non deve essere modificata o a cui si fa riferimento direttamente poiché il suo formato non è fisso e potrebbe cambiare.

Invece, è necessario utilizzare i comandi seguenti per interrogare e modificare i valori nel Registro di sistema:

`dspmqinst` per visualizzare le voci.

`setmqinst` per impostare le voci.

Su Windows, i comandi **`crtmqinst`** e **`dltmqinst`** non sono disponibili. I processi di installazione e disinstallazione gestiscono la creazione e l'eliminazione delle voci di registro richieste.

Multi

IBM MQ MQI client file di configurazione, `mqclient.ini`

È possibile configurare i client utilizzando gli attributi in un file di testo. Questi attributi possono essere sovrascritti dalle variabili di ambiente o in altri modi specifici della piattaforma.

Configurare IBM MQ MQI clients utilizzando un file di testo, simile al file di configurazione del gestore code, `qm.ini`. Il file contiene un numero di stanze, ognuna delle quali contiene un numero di righe del formato **`attribute-name = valore`**.

Il file di configurazione IBM MQ MQI client è generalmente denominato `mqclient.ini`, ma è possibile scegliere di assegnarvi un altro nome. Le informazioni di configurazione in questo file si applicano alle seguenti piattaforme:

- ▶ **ALW** AIX, Linux, and Windows
- ▶ **IBM i** IBM i

Nota: Su IBM i, non esiste alcun file `mqclient.ini` predefinito. Tuttavia, è possibile creare il file nell'IFS (Integrated File System) IBM i Integrated File System.

Per ulteriori informazioni, consultare [“Ubicazione del file di configurazione client”](#) a pagina 170.

Nota: ▶ **z/OS** La piattaforma z/OS non può essere utilizzata per eseguire client IBM MQ. Pertanto, il file `mqclient.ini` non è presente su IBM MQ for z/OS.

Gli attributi nel file di configurazione IBM MQ MQI client si applicano ai client che utilizzano:

- L'MQI
- IBM MQ classes for Java
- IBM MQ classes for JMS
- IBM MQ classes for .NET
- XMS

Anche se gli attributi nel file di configurazione IBM MQ MQI client si applicano alla maggior parte dei client IBM MQ , esistono alcuni attributi che non vengono letti dai client .NET e XMS .NET gestiti o dai client che utilizzano IBM MQ classes for Java o IBM MQ classes for JMS. Per ulteriori informazioni, consultare [“Quali client IBM MQ possono leggere ciascun attributo”](#) a pagina 171.

Le funzioni di configurazione si applicano a tutte le connessioni che un'applicazione client effettua a qualsiasi gestore code, piuttosto che essere specifiche di una singola connessione a un gestore code. Gli attributi relativi a una connessione a un singolo gestore code possono essere configurati in modo programmatico, ad esempio utilizzando una struttura MQCD o utilizzando una CCDT (Client Channel Definition Table).

Di seguito è riportato un esempio di un file di configurazione client:

```

##* Module Name: mqclient.ini                                *#
##* Type       : IBM MQ MQI client configuration file       *#
## Function    : Define the configuration of a client      *#
##*           *#
##*****#
##* Notes      :                                           *#
##* 1) This file defines the configuration of a client     *#
##*           *#
##*****#

ClientExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

TCP:
  Library1=DLLName1
  KeepAlive = Yes
  ClntSndBuffSize=32768
  ClntRcvBuffSize=32768
  Connect_Timeout=0

MessageBuffer:
  MaximumSize=-1
  Updatepercentage=-1
  PurgeTime=0

LU62:
  TPName
  Library1=DLLName1
  Library2=DLLName2

PreConnect:
  Module=myMod
  Function=myFunc
  Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
  Sequence=1

CHANNELS:
  DefRecon=YES
  ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)

Connection:
  ApplName=ExampleApplName

```

Non è possibile impostare più connessioni di canale utilizzando il file di configurazione client.

Le variabili di ambiente supportate nelle release precedenti a IBM WebSphere MQ 7.0 continuano ad essere supportate nelle release successive e, se tale variabile di ambiente corrisponde a un valore equivalente nel file di configurazione del client, la variabile di ambiente sovrascrive il valore del file di configurazione del client.

Per un'applicazione client che utilizza IBM MQ classes for JMS, è anche possibile sovrascrivere il file di configurazione client nei modi seguenti:

- Impostando le proprietà nel file di configurazione JMS .
- Impostando le proprietà di sistema Java , che sovrascrive anche il file di configurazione JMS .

Per il client .NET , è possibile anche sovrascrivere il file di configurazione client e le variabili di ambiente equivalenti utilizzando il file di configurazione dell'applicazione .NET .

Commenti nel file di configurazione

Linux

AIX

È possibile utilizzare il punto e virgola ';' e l'hash '#' per contrassegnare l'inizio di un commento all'interno del file di configurazione. Ciò può contrassegnare un'intera riga come commento o indicare un commento alla fine di una riga che non verrà incluso nel valore di un'impostazione.

Se un valore richiede uno di questi caratteri, è necessario eseguire l'escape di tale carattere utilizzando il carattere barra rovesciata '\\'.

Il seguente esempio mostra l'utilizzo dei commenti nel file di configurazione:

```
# Example of an SSL stanza with comments
SSL:
  ClientRevocationChecks=REQUIRED ; Example of an end of line comment
  SSLCryptoHardware=GSK_PKCS11=/driver\;label\;password\;SYMMETRIC_CIPHER_ON # Example of
  escaped comment characters.
```

Concetti correlati

[Il file di configurazione delle classi IBM MQ per Java](#)

Multi

Ubicazione del file di configurazione client

Un file di configurazione IBM MQ MQI client può essere contenuto in un numero di ubicazioni.

Un'applicazione del client utilizza il seguente percorso di ricerca per individuare il file di configurazione IBM MQ MQI client :

1. L'ubicazione specificata dalla variabile di ambiente **MQCLNTCF**.

Il formato di questa variabile di ambiente è un URL completo. Ciò significa che il nome file potrebbe non essere necessariamente `mqclient.ini` e facilita l'inserimento del file su un file system collegato in rete.

Note:

- I client C, .NET e XMS supportano solo il protocollo `file:` ; il protocollo `file:` viene utilizzato se la stringa URL non inizia con `protocol:`
 - Per consentire JRE Java 1.4.2 , che non supportano la lettura di variabili di ambiente, la variabile di ambiente **MQCLNTCF** può essere sovrascritta con una proprietà di sistema **MQCLNTCF** Java .
2. Un file denominato `mqclient.ini` nella directory di lavoro corrente dell'applicazione.
 3. Un file denominato `mqclient.ini` nella directory di dati IBM MQ per sistemi AIX, Linux, and Windows .

Note:

- La directory di dati IBM MQ non esiste nei casi seguenti:

–  AttivoIBM i

–  Attivoz/OS

– dove il client è stato fornito con un altro prodotto

 Su IBM i, non esiste alcun file `mqclient.ini` predefinito. Tuttavia, il file può essere creato nell'IFS (Integrated File System) IBM i Integrated File System nell'indirizzario `/QIBM/UserData/mqm/e` la variabile di ambiente **MQCLNTCF** definita per puntare ad esso. Ad esempio:

```
ADDENVVAR ENVVAR(MQCLNTCF) VALUE('QIBM/UserData/mqm/mqclient.ini') REPLACE(*YES)
```

Per ulteriori esempi di variabili di ambiente, consultare [“Descrizioni delle variabili di ambiente” a pagina 66](#).

► **z/OS** La piattaforma z/OS non può essere utilizzata per eseguire client IBM MQ . Pertanto, il file `mqclient.ini` non è presente su IBM MQ for z/OS.

- **Linux** ► **AIX** Su sistemi AIX and Linux , la directory è `/var/mqm`.
- **Windows** Su piattaforme Windows , configurare la variabile di ambiente `MQ_DATA_PATH` durante l'installazione in modo che punti alla directory dei dati. Di solito è `C:\ProgramData\IBM\MQ`.

Nota: Se si sta installando solo un client, la variabile di ambiente potrebbe essere `MQ_FILE_PATH`.

- Per consentire i JRE Java 1.4.2 che non supportano la lettura di variabili di ambiente, è possibile sovrascrivere manualmente la variabile di ambiente `MQ_DATA_PATH` con una proprietà di sistema `MQ_DATA_PATH Java` .
4. Un file denominato `mqclient.ini` in una directory standard appropriata per la piattaforma e accessibile agli utenti:
- Per tutti i client Java questo è il valore della proprietà di sistema `user.home Java` .
 - **Linux** ► **AIX** Per client C su piattaforme AIX and Linux questo è il valore della variabile di ambiente `HOME` .
 - **Windows** Per client C su Windows , sono i valori concatenati delle variabili di ambiente `HOMEDRIVE` e `HOMEPATH` .

► **Multi** Quali client IBM MQ possono leggere ciascun attributo

La maggior parte degli attributi nel file di configurazione IBM MQ MQI client può essere utilizzata dal client C e dai client .NET non gestiti. Tuttavia, esistono alcuni attributi che non vengono letti dai client .NET e XMS .NET gestiti o dai client che utilizzano IBM MQ classes for Java o IBM MQ classes for JMS.

<i>Tabella 16. Quali attributi si applicano a ciascun tipo di client</i>						
Nome e attributi della stanza <code>mqclient.ini</code>	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
Stanza CHANNELS						
<u>CCSID</u>	Il numero della serie di caratteri codificati da utilizzare.	Sì	No	No	Sì	Sì
<u>ChannelDefinitionDirectory</u>	Il percorso della directory del file contenente la tabella di definizione del canale client.	Sì	No	No	Sì	Sì

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>File ChannelDefinition</u>	Il nome del file contenente la tabella di definizione del canale client.	Sì	No	No	Sì	Sì
<u>ReconDelay</u>	Un'opzione di gestione per configurare il ritardo di riconnession e per i programmi client che possono riconnettersi automaticamente.	Sì	No	Sì	Sì	Sì
<u>DefRecon</u>	Un'opzione di gestione per abilitare i programmi client a riconnettersi automaticamente o per disabilitare la riconnession e automatica di un programma client che è stato scritto per riconnettersi automaticamente.	Sì	No	Sì	Sì	Sì
<u>MQReconnectTimeout</u>	Il timeout in secondi per riconnettersi a un client.	Sì	No	No	Sì	No

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza <code>mqclient.ini</code>	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
ServerConnectionParameters	L'ubicazione del server IBM MQ e il metodo di comunicazione da utilizzare.	Sì	No	No	Sì	Sì
Put1DefaultAlwaysSync	Controlla il comportamento della chiamata della funzione MQPUT1 con opzione MQPMO_RESPONSE_AS_Q_DEF.	Sì	Sì	Sì	Sì	Sì
PasswordProtection	Consente di impostare password protette nella struttura MQCSP, piuttosto che utilizzare SSL o TLS.	Sì	Sì	Sì	Sì	Sì
StanzaClientExitPath						
ExitsDefaultPath	Specifica l'ubicazione delle uscite canale a 32 bit per i client.	Sì	Sì	Sì	Sì	Sì
ExitsDefaultPath64	Specifica l'ubicazione delle uscite del canali a 64 bit per i client.	Sì	Sì	Sì	Sì	Sì

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza <code>mqclient.ini</code>	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>JavaExitsClassPath</u>	I valori da aggiungere al percorso classi quando viene eseguita un'uscita Java .	No	Sì	Sì	No	No
Stanza Connessione						
<u>ApplName</u>	Il nome dell'applicazione specificato nel file di configurazione e client.	Sì	No	No	No	No
Stanza JMQUI						
<u>useMQCSPAuthentication</u>	Controlla se le applicazioni di IBM MQ classes for Java e IBM MQ classes for JMS devono utilizzare la modalità di compatibilità o la modalità di autenticazione e MQCSP durante l'autenticazione con un gestore code.	No	Sì	Sì	No	No
Stanza MessageBuffer						

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>MaximumSize</u>	La dimensione, in kilobyte, del buffer di lettura anticipata, nell'intervallo compreso tra 1 e 999 999.	Sì	Sì	Sì	Sì	Sì
<u>PurgeTime</u>	Intervallo, in secondi, dopo il quale vengono eliminati i messaggi rimasti nel buffer di lettura anticipata.	Sì	Sì	Sì	Sì	Sì
<u>UpdatePercentage</u>	Il valore percentuale di aggiornamento, nell'intervallo compreso tra 1 e 100, utilizzato per calcolare il valore di soglia per stabilire quando un'applicazione client effettua una nuova richiesta al server.	Sì	Sì	Sì	Sì	Sì
Stanza PreConnect						

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>Dati</u>	URL del repository in cui sono memorizzate le definizioni di connessione.	Sì	No	No	No	No
<u>funzione</u>	Nome del punto di immissione funzionale nella libreria che contiene il codice di uscita PreConnect .	Sì	No	No	No	No
<u>Modulo</u>	Il nome del modulo contenente il codice di uscita API.	Sì	No	No	No	No
<u>Sequenza</u>	La sequenza in cui questa uscita viene chiamata rispetto ad altre uscite.	Sì	No	No	No	No
Stanza di sicurezza						
<u>DisableClient</u> <u>DisableClient</u>	Disabilita o abilita AMS per le connessioni client a un gestore code.	Sì	Sì	Sì	No	No
Stanza SSL						

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)



Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>OutboundSNI</u>	Specifica se i client con capacità SNI imposteranno SNI sul nome del canale IBM MQ di destinazione sul sistema remoto durante l'avvio di una connessione TLS o sul nome host.	Sì	Sì	Sì	Sì	No
<u>AllowOutboundAllowOutbound</u>	<p>Specifica se i client con capacità SNI imposteranno SNI sul nome canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS.</p> <p> Attenzione:  Deprecated Da IBM MQ 9.3.0 questa proprietà è obsoleta. Utilizzare invece OutboundSNI.</p>	Sì	Sì	Sì	No	No
<u>AllowTLSV13</u>	Se un gestore code è in grado di utilizzare TLS 1.3 CipherSpecs.	Sì (client C/C++)	No	No	No	No

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
CDPCheckExtensions	Specifica se i canali SSL o TLS su questo gestore code tentano di controllare i server CDP denominati nelle estensioni certificato del punto CrlDistribution.	Sì	No	No	No	No
CertificateLabel	L'etichetta del certificato della definizione del canale.	Sì	No	No	No	No
Politica CertificateValidation	Determina il tipo di convalida del certificato utilizzato.	Sì	No	No	No	No
ClientRevocationClientRevocation	Determina in che modo è configurato il controllo della revoca del certificato se la chiamata di connessione client utilizza un canale SSL/TLS.	Sì	No	No	No	No

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>EncryptionPolicySuiteB</u>	Determina se un canale utilizza la crittografia conforme a Suite - B e quale livello di potenza deve essere utilizzato.	Sì	No	No	No	No
<u>EnvironmentScope</u>	Controlla se IBM MQ utilizza un singolo ambiente IBM Global Security Kit (GSKit) per l'intero processo o un ambiente GSKit per connessione.	Sì (client C)	No	No	No	No
<u>MinimumRSAKeyDimensione</u>	Specifica la dimensione chiave minima che i certificati RSA devono avere per essere accettati.	Sì (client C/C++)	No	No	No	No
<u>OCSPAuthentication</u>	Definisce il comportamento di IBM MQ quando OCSP è abilitato e il controllo della revoca OCSP non è in grado di determinare lo stato della revoca del certificato.	Sì	No	No	No	No

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)


Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>OCSPCheckExtensions</u>	Controlla se IBM MQ agisce sulle estensioni del certificato di accesso AuthorityInfo .	Sì	No	No	No	No
<u>OCSPTimeout</u>	Il numero di secondi di attesa per un responder OCSP durante l'esecuzione di una verifica di revoca.	Sì	No	No	No	No
 <u>PeerCertChainValidation</u>	L'impostazione di convalida del certificato GSKit .	Sì	No	No	No	No
<u>SSLCryptoHardware</u>	Imposta la stringa del parametro richiesta per configurare l'hardware crittografico PKCS #11 presente sul sistema.	Sì	No	No	No	No

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
SSLCryptoHardwareKeyFile	Specifica il percorso completo e il nome del file contenente la chiave iniziale utilizzata per codificare la password nella stringa di configurazione e hardware crittografica PKCS #11 specificata con l'attributo SSLCryptoHardware .	Sì	No	No	No	No
SSLFipsRequired	Specifica se devono essere utilizzati solo algoritmi certificati FIPS se la crittografia viene eseguita in IBM MQ.	Sì	No	No	No	No
SSLHTTPProxyName	La stringa è il nome host o l'indirizzo di rete del server proxy HTTP che deve essere utilizzato da GSKit per i controlli OCSP.	Sì	No	No	No	No

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>SSLHTTPConnectTimeout</u>	Il numero di secondi di attesa per una connessione di rete stabilita correttamente e con un server HTTP durante l'esecuzione di un controllo di revoca.	Sì	No	No	No	No
<u>SSLKeyRepository</u>	L'ubicazione del repository di chiavi che contiene il certificato digitale dell'utente, in formato stem.	Sì	No	No	No	No
<u>SSLKeyRepositoryPassword</u>	La passphrase per accedere al repository delle chiavi.	Sì	No	No	No	No
<u>SSLKeyResetConteggio</u>	Il numero di byte non crittografati inviati e ricevuti su un canale SSL o TLS prima che la chiave segreta venga rinegoziata.	Sì	No	No	No	No
Stanza TCP						

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
<u>ClntRcvBufferSize</u>	La dimensione, in byte, del buffer di ricezione TCP/IP utilizzato dall'estremità client di un canale di connessione server di connessione client.	Sì	Sì	Sì	Sì	Sì
<u>ClntSndBufferSize</u>	La dimensione in byte del buffer di invio TCP/IP utilizzato dall'estremità client di un canale di connessione server di connessione client.	Sì	Sì	Sì	Sì	Sì
<u>Timeout connessione</u>	Il numero di secondi prima del timeout di un tentativo di connessione del socket.	Sì	Sì	Sì	No	No
<u>IPAddressVersion</u>	Specifica quale protocollo IP utilizzare per una connessione di canale.	Sì	No	No	Sì	Sì
<u>KeepAlive</u>	Attiva o disattiva la funzione KeepAlive .	Sì	Sì	Sì	Sì	Sì

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
► Windows Library1	Solo su Windows , il nome della DLL dei socket TCP/IP.	Sì	No	No	No	No
Stanza di traccia						
Nota: La stanza Trace si riferisce solo ai client IBM MQ .NET e XMS .NET .						
► V 9.4.0 Livello MQDotnetTrace	Utilizzato per abilitare la traccia IBM MQ .NET .	No	No	No	Sì	No
► V 9.4.0 PercorsoMQDotnetTrace	Fa riferimento a una cartella in cui verranno creati i file di traccia di IBM MQ .NET .	No	No	No	Sì	No
► V 9.4.0 MQDotnetErrorMQDotnetError	Indica una cartella in cui verranno creati i file di log degli errori per la traccia di IBM MQ .NET .	No	No	No	Sì	No
► V 9.4.0 LivelloXMSDotnetTrace	Utilizzato per abilitare la traccia XMS .NET .	No	No	No	No	Sì
► V 9.4.0 XMSDotnetTraceFilePath	Punta ad una cartella in cui verranno creati i file di traccia XMS .NET .	No	No	No	No	Sì

Tabella 16. Quali attributi si applicano a ciascun tipo di client (Continua)

Nome e attributi della stanza mqclient.ini	Descrizione	C e .NET non gestito	Java	JMS	Gestita.NET	GestitaXMS .NET
> V9.4.0 SpecificaXMSDotnetTrace	Specifica il nome della classe che si desidera tracciare per XMS .NET.	No	No	No	No	Sì
> V9.4.0 SpecificaXMSDotnetTrace	Specifica la dimensione massima del file di traccia che deve essere generato per XMS .NET.	No	No	No	No	Sì
> V9.4.0 XMSDotnetTraceFileSize	Il numero di file di traccia da conservare per XMS .NET.	No	No	No	No	Sì

Stanza Application del file di configurazione client

Utilizzare la sezione Application per specificare attributi che influenzano il funzionamento del bilanciamento del cluster uniforme per una specifica applicazione che si connette utilizzando questa configurazione. I valori in questa stanza hanno la precedenza sulla stanza ApplicationDefaults, ma possono essere sovrascritti da una struttura MQBNO fornita tramite un programma.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella stanza Application:

Nome = ApplicationName

Identifica a quale nome applicazione si applicano le opzioni.

Tipo = *Semplice,ReqRep*

Indica IBM MQ il pattern generale dell'attività IBM MQ a cui partecipa questa applicazione.

BalanceTimeout = *Never,Immediate,0-999999999,predefinito*

Indica a IBM MQ il timeout prima che l'attività dell'applicazione possa essere interrotta per consentire il ribilanciamento; mai o un valore fino a un massimo di 999.999.999 secondi, con un valore predefinito di 10 secondi.

BalanceOptions = *Nessuno,IgnTrans*

Non è impostata alcuna opzione di bilanciamento o consente l'interruzione immediata delle applicazioni attualmente coinvolte in una transazione.

Stanza ApplicationDefaults del file di configurazione client

Utilizzare la sezione ApplicationDefaults per specificare attributi che influenzano il comportamento di bilanciamento del cluster uniforme predefinito per applicazioni client che si collegano utilizzando questa configurazione. Questi valori predefiniti possono essere sovrascritti da una sezione Application specifica dell'applicazione o da una struttura MQBNO fornita tramite un programma.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella sezione ApplicationDefaults :

Tipo = *Semplice,ReqRep*

Indica IBM MQ il pattern generale dell'attività IBM MQ a cui partecipa questa applicazione.

BalanceTimeout = *Never,Immediate,0-999999999, predefinito*

Indica a IBM MQ il timeout prima che l'attività dell'applicazione possa essere interrotta per consentire il ribilanciamento; mai o un valore fino a un massimo di 999.999.999 secondi, con un valore predefinito di 10 secondi.

BalanceOptions = *Nessuno,IgnTrans*

Non è impostata alcuna opzione di bilanciamento o consente l'interruzione immediata delle applicazioni attualmente coinvolte in una transazione.

Stanza dei canali del file di configurazione client

Utilizzare la stanza Canali per specificare le informazioni sui canali client.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella stanza Canali:

CCSID = *numero*

Il numero della serie di caratteri codificati da utilizzare.

Questo attributo può essere letto da client C, .NETnon gestiti, .NETgestiti e XMS .NET gestiti.

Il numero CCSID è equivalente alla variabile di ambiente [MQCCSID](#) .

ChannelDefinitionDirectory = *percorso*

Il percorso della directory del file contenente la tabella di definizione del canale client.

Questo attributo può essere letto da client C, .NETnon gestiti, .NETgestiti e XMS .NET gestiti.

Windows Sui sistemi Windows , il valore predefinito è la directory dei file di log e di dati IBM MQ , generalmente C : \ProgramData\IBM\MQ.

Linux **AIX** Su sistemi AIX and Linux , il valore predefinito è /var/mqm.

La directory ChannelDefinition può contenere un URL che funziona insieme all'attributo File ChannelDefinition (consultare ["Accesso URL alla CCDT"](#) a pagina 54).

Il percorso della directory ChannelDefinition è equivalente alla variabile di ambiente [MQCHLLIB](#) .

ChannelDefinitionFile = *nomefile|AMQCLCHL . TAB*

Il nome del file contenente la tabella di definizione del canale client.

Questo attributo può essere letto da client C, .NETnon gestiti, .NETgestiti e XMS .NET gestiti.

La tabella di definizione del canale client è equivalente alla variabile di ambiente [MQCHLTAB](#) .

ReconDelay = (*ritardo [, rand]*) (*ritardo [, rand]*) ...

L'attributo ReconDelay fornisce un'opzione di gestione per configurare il ritardo di riconnessione per i programmi client che possono riconnettersi automaticamente.

Questo attributo può essere letto da client C, .NET non gestiti, IBM MQ classes for JMS, .NET gestiti e XMS .NET gestiti.

Ecco un esempio di configurazione:

```
ReconDelay=(1000,200) (2000,200) (4000,1000)
```

L'esempio mostrato definisce un ritardo iniziale di un secondo, più un intervallo casuale fino a 200 millisecondi. Il ritardo successivo è di due secondi più un intervallo casuale di un massimo di 200 millisecondi. Tutti i ritardi successivi sono di quattro secondi, più un intervallo casuale fino a 1000 millisecondi.

DefRecon = NO|YES|QMGR |DISABLED

L'attributo DefRecon fornisce un'opzione di gestione per abilitare i programmi client a riconnettersi automaticamente o per disabilitare la riconnessione automatica di un programma client che è stato scritto per riconnettersi automaticamente. È possibile scegliere di impostare quest'ultima opzione se un programma utilizza un'opzione, come MQPMO_LOGICAL_ORDER, che non è compatibile con la riconnessione.

Questo attributo può essere letto da client C, .NET non gestiti, IBM MQ classes for JMS, .NET gestiti e XMS .NET gestiti .

La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

La riconnessione automatica del client generalmente dipende da due valori che sono:

- Opzione di riconnessione impostata nell'applicazione MQCONNX (o factory di connessione JMS)
- Opzione di riconnessione predefinita fornita in qualsiasi definizione di connessione client in uso (struttura MQCD, ad esempio fornita utilizzando un file CCDT).

L'attributo del file mqclient.ini si applica **solo** se non viene utilizzata alcuna definizione di canale che imposta l'attributo **DefReconnect** e in tale situazione si comporta come se ne fosse stata fornita una. L'attributo **DefReconnect** del canale (e quindi questo attributo, se applicabile):

- Sovrascrivere il codice dell'applicazione se è impostato su DISABLED
- Vengono sovrascritti dal codice dell'applicazione in tutti gli altri casi, se le opzioni sono specificate su MQCONNX

Consultare la descrizione di [DEFRECON](#) per una tabella che mostra tutte le possibili combinazioni di valori forniti per l'applicazione e la definizione del canale.

Note:

- Se un MQCD è in uso ma precede MQCD_VERSION_10, il parametro **DefReconnect** non fa parte della struttura. In questa situazione, il valore di tale parametro mancante viene popolato con il valore mqclient.ini file **DefReconnect** , se specificato. Ciò può verificarsi, ad esempio, se un formato binario CCDT generato in una versione precedente di IBM MQ è ancora utilizzato da un'applicazione client.
- Quando interpretato dal codice client IBM MQ , una CCDT JSON, consultare [“Configurazione di un formato JSON CCDT”](#) a pagina 46, genera sempre le strutture MQCD alla versione più recente e quindi fornisce sempre il valore predefinito (NO) per questo attributo a meno che non sia esplicitamente presente con un valore diverso.

MQReconnectTimeout

Il tempo massimo, in secondi, in cui la funzione di riconnessione automatica del client in un client tenta di ristabilire la connessione. Il valore predefinito è 1800 secondi (30 minuti).

Questo attributo può essere letto da client C e .NET non gestiti e da client .NET gestiti.

I client IBM MQ classes for JMS possono specificare un timeout per riconnettersi utilizzando la proprietà del factory di connessione [CLIENTRECONNECTTIMEOUT](#). Il valore predefinito per questa proprietà è 1800 secondi (30 minuti).

I client IBM MQ classes for XMS .NET possono specificare un timeout per riconnettersi utilizzando le seguenti proprietà:

- La proprietà factory di connessione CLIENTRECONNECTTIMEOUT. Il valore predefinito per questa proprietà è 1800 secondi (30 minuti). Questa proprietà è valida solo per la modalità gestita.
- La proprietà XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT. Il valore predefinito per questa proprietà è 1800 secondi (30 minuti). Questa proprietà è valida solo per la modalità gestita.

Parametri ServerConnection

ServerConnectionParms è equivalente alla variabile di ambiente MQSERVER e specifica l'ubicazione del server IBM MQ e il metodo di comunicazione da utilizzare.

Questo attributo può essere letto da client C, .NETnon gestiti, .NETgestiti e XMS .NET gestiti.

L'attributo parametri ServerConnectiondefinisce solo un canale semplice; non è possibile utilizzarlo per definire un canale TLS o un canale con uscite canale. È una stringa nel formato *ChannelName/TransportType/ConnectionName, ConnectionName* deve essere un nome di rete completo. *ChannelName* non può contenere il carattere barra (/) poiché questo carattere è utilizzato per separare il nome del canale, il tipo di trasporto e il nome della connessione.

Quando si utilizzano i parametri ServerConnectionper definire un canale client, viene utilizzata una lunghezza massima di 100 MB. Pertanto, la dimensione massima del messaggio in vigore per il canale è il valore specificato nel canale SVRCONN sul server.

Tenere presente che è possibile effettuare una singola connessione del canale client. Ad esempio, se si dispone di due voci:

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

viene utilizzato solo il secondo.

Specificare *ConnectionName* come un elenco separato da virgole di nomi per il tipo di trasporto indicato. In genere, è richiesto un solo nome. È possibile fornire più *nomi host* per configurare più connessioni con le stesse proprietà. Le connessioni vengono tentate nell'ordine in cui sono specificate nell'elenco delle connessioni fino a quando non viene stabilita correttamente una connessione. Se nessuna connessione ha esito positivo, il client inizia nuovamente l'elaborazione. Gli elenchi di connessioni sono un'alternativa ai gruppi di gestori code per configurare le connessioni per i client ricollegabili.

Put1DefaultAlwaysSync = NO (predefinito) | YES

Controlla il comportamento della chiamata della funzione MQPUT1 con opzione MQPMO_RESPONSE_AS_Q_DEF.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

No

Se MQPUT1 è impostato con MQPMO_SYNCPOINT, si comporta come MQPMO_ASYNC_RESPONSE. Allo stesso modo, se MQPUT1 è impostato con MQPMO_NO_SYNCPOINT, si comporta come MQPMO_SYNC_RESPONSE. Questo è il valore predefinito.

Sì

MQPUT1 si comporta come se MQPMO_SYNC_RESPONSE fosse impostato, indipendentemente dal fatto che MQPMO_SYNCPOINT o MQPMO_NO_SYNCPOINT sia impostato.

PasswordProtection = Compatibile (predefinito) |always|facoltativo

Da IBM MQ 8.0, le credenziali di autenticazione che le applicazioni IBM MQ client specificano quando si collegano a un gestore code possono essere protette utilizzando la funzione di protezione della password MQCSP IBM MQ, se la connessione non utilizza la codifica TLS.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

La protezione della password MQCSP è utile per scopi di test e sviluppo in quanto l'utilizzo della protezione della password MQCSP è più semplice rispetto all'impostazione della crittografia TLS, ma non così sicuro.

Per ulteriori informazioni sulla protezione delle credenziali nella struttura MQCSP e sui valori che possono essere impostati per questo attributo, consultare [Protezione password MQCSP](#).

Attività correlate

[Connessione delle applicazioni MQI IBM MQ ai gestori code](#)

Multi

Stanza di percorso ClientExit del file di configurazione del client

Utilizzare la stanza ClientExitPath per specificare le ubicazioni predefinite delle uscite del canale sul client.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella stanza Percorso ClientExit:

ExitsDefaultPercorso = *stringa*

Specifica l'ubicazione delle uscite di canale a 32 bit per i clienti.

Questo attributo può essere letto da client C, .NET non gestiti, .NET gestiti, XMS .NET gestiti, IBM MQ classes for Java e IBM MQ classes for JMS. I client IBM MQ classes for Java e IBM MQ classes for JMS utilizzano questo attributo per individuare le uscite di canale a 32 bit non scritte in Java.

ExitsDefaultPath64 = *stringa*

Specifica l'ubicazione delle uscite del canali a 64 bit per i client.

Questo attributo può essere letto da client C, .NET non gestiti, .NET gestiti, XMS .NET gestiti, IBM MQ classes for Java e IBM MQ classes for JMS. I client IBM MQ classes for Java e IBM MQ classes for JMS utilizzano questo attributo per individuare le uscite del canale a 64 bit non scritte in Java.

JavaExitsClassPath = *stringa*

I valori da aggiungere al percorso classi quando viene eseguita un'uscita Java. Ciò viene ignorato dalle uscite in qualsiasi altra lingua.

Questo attributo può essere letto da client IBM MQ classes for Java e IBM MQ classes for JMS.

Nel file di configurazione JMS, al nome percorso JavaExitsClass viene fornito lo standard com.ibm.mq.cfg. e questo nome completo viene utilizzato anche sulla proprietà di sistema IBM MQ.

Multi

Stanza di connessione del file di configurazione client

Utilizzare la stanza di connessione per specificare un nome applicazione.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

Il seguente attributo può essere incluso nella stanza Connection:

ApplName = *ExampleApp1name*

È possibile specificare un nome applicazione nel file di configurazione client.

Questo attributo può essere utilizzato da client C e .NET non gestiti.

Multi

Stanza JMQUI del file di configurazione client

Utilizzare la stanza JMQUI per specificare i parametri di configurazione per JMQUI (Java Message Queuing Interface) utilizzati da IBM MQ classes for Java e IBM MQ classes for JMS.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

Il seguente attributo può essere incluso nella stanza JMQUI:

useMQCSPauthentication = NO|SI

Controlla se le applicazioni di IBM MQ classes for Java e IBM MQ classes for JMS devono utilizzare la modalità di compatibilità o la modalità di autenticazione MQCSP durante l'autenticazione con un gestore code.

Questo attributo può essere letto da client IBM MQ classes for Java e IBM MQ classes for JMS .

Questo attributo può avere i seguenti valori:

No

Utilizzare la modalità di compatibilità durante l'autenticazione con un gestore code. Questo è il valore predefinito nelle versioni precedenti a IBM MQ 9.3.0.

si

Utilizzare la modalità di autenticazione MQCSP durante l'autenticazione con un gestore code. Questo è il valore predefinito da IBM MQ 9.3.0.

Esistono diversi altri modi per impostare la modalità di autenticazione che hanno la precedenza sul valore dell'attributo **useMQCSPauthentication** . Per ulteriori informazioni sulla modalità di compatibilità e la modalità di autenticazione MQCSP, consultare [Autenticazione della connessione con il client Java](#).



LU62, NETBIOS e stanze SPX del file di configurazione client

Solo su sistemi Windows , utilizzare queste stanze per specificare i parametri di configurazione per i protocolli di rete specificati.

stanza LU62

Utilizzare la stanza LU62 per specificare i parametri di configurazione del protocollo SNA LU 6.2 . I seguenti attributi possono essere inclusi in questa sezione:

Library1 = DLLName|WCPIC32

Il nome della DLL APPC.

Library2 = DLLName|WCPIC32

Lo stesso di Library1, utilizzato se il codice è memorizzato in due librerie separate.

TPName

Il nome TP da avviare sul sito remoto.

stanza NETBIOS

Utilizzare la stanza NETBIOS per specificare i parametri di configurazione del protocollo NetBIOS . I seguenti attributi possono essere inclusi in questa sezione:

AdapterNum = numero|0

Il numero dell'adattatore LAN.

Library1 = NomeDLL|NETAPI32

Il nome della DLL NetBIOS.

LocalName = nome

Il nome con cui questo computer è noto sulla LAN.

È equivalente alla variabile di ambiente [MQNAME](#) .

NumCmds = numero|1

Quanti comandi allocare.

NumSess = numero|1

Quante sessioni allocare.

Stanza SPX

Utilizzare la stanza SPX per specificare i parametri di configurazione del protocollo SPX. I seguenti attributi possono essere inclusi in questa sezione:

BoardNum = numero|0

Il numero dell'adattatore LAN.

KeepAlive = SÌ|NO

Attivare o disattivare la funzione KeepAlive .

KeepAlive = YES fa sì che SPX controlli periodicamente che l'altra estremità della connessione sia ancora disponibile. In caso contrario, il canale viene chiuso.

Library1 = DLLName|WSOCK32 . DLL

Il nome della DLL SPX.

Library2 = NomeDLL|WSOCK32 . DLL

Lo stesso di Library1, utilizzato se il codice è memorizzato in due librerie separate.

Socket = numero|5E86

Il numero di socket SPX in notazione esadecimale.

Stanza MessageBuffer del file di configurazione client

Utilizzare la sezione MessageBuffer per specificare informazioni sui buffer di messaggi.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella sezione MessageBuffer :

MaximumSize = numero intero|1

La dimensione, in kilobyte, del buffer di lettura anticipata, nell'intervallo compreso tra 1 e 999 999.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

Esistono i seguenti valori speciali:

-1

Il client determina il valore appropriato.

0

La lettura anticipata è disabilitata per il client.

PurgeTime = numero intero|600

Intervallo, in secondi, dopo il quale vengono eliminati i messaggi rimasti nel buffer di lettura anticipata.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

Se l'applicazione client sta selezionando i messaggi basati su MsgId o CorrelId , è possibile che il buffer di lettura anticipata contenga messaggi inviati al client con un MsgId o un CorrelIdprecedentemente richiesto. Questi messaggi vengono bloccati nel buffer di lettura anticipata fino a quando non viene emesso un MQGET con un MsgId o CorrelIdappropriato. È possibile eliminare i messaggi dal buffer di lettura anticipata impostando PurgeTime. Tutti i messaggi rimasti nel buffer di lettura anticipata per un periodo più lungo dell'intervallo di eliminazione vengono eliminati automaticamente. Questi messaggi sono già stati rimossi dalla coda sul gestore code, quindi, a meno che non vengano consultati, vengono persi.

L'intervallo valido è compreso tra 1 e 999 999 999 secondi o il valore speciale 0, che indica che non viene eseguita alcuna eliminazione.

UpdatePercentage = numero intero| -1

Il valore percentuale di aggiornamento, nell'intervallo compreso tra 1 e 100, utilizzato per calcolare il valore di soglia per stabilire quando un'applicazione client effettua una nuova richiesta al server. Il valore speciale -1 indica che il client determina il valore appropriato.

Questo attributo può essere letto da client C, .NET non gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NET gestiti e XMS .NET gestiti.

Il client invia periodicamente una richiesta al server indicando la quantità di dati che l'applicazione client ha utilizzato. Una richiesta viene inviata quando il numero di byte, n , richiamati dal client tramite chiamate MQGET supera una soglia T . n viene reimpostato su zero ogni volta che viene inviata una nuova richiesta al server.

La soglia T è calcolata come segue:

$$T = Upper - Lower$$

Il valore superiore è uguale alla dimensione del buffer di lettura anticipata, specificata dall'attributo *MaximumSize*, in kilobyte. Il suo valore predefinito è 100 Kb.

Inferiore è inferiore a Superiore ed è specificato dall'attributo *UpdatePercentage*. Questo attributo è un numero compreso nell'intervallo tra 1 e 100 e ha un valore predefinito di 20. Inferiore è calcolato come segue:

$$Lower = Upper \times UpdatePercentage / 100$$

Esempio 1:

Gli attributi *MaximumSize* e *UpdatePercentage* assumono i valori predefiniti di 100 Kb e 20 Kb.

Il client richiama MQGET per richiamare un messaggio e lo fa ripetutamente. Questa operazione continua fino a quando MQGET non ha utilizzato n byte.

Utilizzo del calcolo

$$T = Upper - Lower$$

T è $(100 - 20) = 80$ Kb.

Quindi, quando le chiamate MQGET hanno rimosso 80 Kb da una coda, il client effettua automaticamente una nuova richiesta.

Esempio 2:

Gli attributi *MaximumSize* assumono il valore predefinito di 100 Kb e viene scelto un valore di 40 per *UpdatePercentage*.

Il client richiama MQGET per richiamare un messaggio e lo fa ripetutamente. Questa operazione continua fino a quando MQGET non ha utilizzato n byte.

Utilizzo del calcolo

$$T = Upper - Lower$$

T è $(100 - 40) = 60$ Kb

Quindi, quando le chiamate MQGET hanno rimosso 60 Kb da una coda, il client effettua automaticamente una nuova richiesta. Ciò è più rapido rispetto all'ESEMPIO 1 in cui sono stati utilizzati i valori predefiniti.

Pertanto, la scelta di una soglia maggiore di T tende a ridurre la frequenza con cui le richieste vengono inviate dal client al server. Al contrario, la scelta di una soglia più piccola T tende ad aumentare la frequenza delle richieste inviate dal client al server.

Tuttavia, la scelta di una soglia elevata T può significare che il guadagno di prestazioni della lettura anticipata viene ridotto man mano che aumenta la possibilità che il buffer di lettura anticipata diventi

vuoto. Quando ciò accade, una chiamata MQGET potrebbe dover essere sospesa, in attesa che i dati arrivino dal server.

Multi

Stanza PreConnect del file di configurazione client

Utilizzare la stanza PreConnect per configurare l'uscita PreConnect nel file `mqclient.ini`.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella stanza PreConnect :

Data = *dati_utente*

Questo attributo specifica i dati utente passati all'uscita di preconnessione. I dati passati all'uscita di preconnessione sono specifici per l'implementazione dell'exit di preconnessione che si sta utilizzando e quali dati si prevede di trasmettere.

Questo attributo può essere letto da client C e .NET non gestiti.

Ad esempio, questo attributo può essere utilizzato per specificare l'URL del repository in cui sono memorizzate le definizioni di connessione, ad esempio, quando si utilizza un server LDAP:

```
Data = ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
```

Funzione = *myFunc*

Nome del punto di immissione funzionale nella libreria che contiene il codice di uscita PreConnect .

Questo attributo può essere letto da client C e .NET non gestiti.

La definizione della funzione aderisce al prototipo di uscita PreConnect [MQ_PRECONNECT_EXIT](#).

La lunghezza massima di questo campo è `MQ_EXIT_NAME_LENGTH`.

Modulo = *myMod*

Il nome del modulo contenente il codice di uscita API.

Questo attributo può essere letto da client C e .NET non gestiti.

Se questo campo contiene il nome percorso completo del modulo, viene utilizzato così com'è.

Sequenza = *numero_sequenza*

La sequenza in cui questa uscita viene chiamata rispetto ad altre uscite. Un'uscita con un numero di sequenza basso viene richiamata prima di un'uscita con un numero di sequenza più alto. Non c'è bisogno che la numerazione di sequenza delle uscite sia continua; una sequenza di 1, 2, 3 ha lo stesso risultato di una sequenza di 7, 42, 1096. Questo attributo è un valore numerico senza segno.

Questo attributo può essere letto da client C e .NET non gestiti.

Più stanze PreConnect possono essere definite all'interno del file `mqclient.ini`. L'ordine di elaborazione di ciascuna uscita è determinato dall'attributo Sequenza della stanza.

Attività correlate

[Riferimento alle definizioni di connessione mediante un'uscita di pre - connessione da un repository](#)

Multi

Stanza di sicurezza del file di configurazione client

Utilizzare la stanza di sicurezza per disattivare o abilitare AMS per le connessioni client a un gestore code.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

Il seguente attributo può essere incluso nella stanza Security:

DisableClientAMS = NO|SI

L'attributo DisableClientAMS consente di disabilitare IBM MQ Advanced Message Security (AMS) se si utilizza un client IBM MQ per connettersi a un gestore code da una versione precedente del prodotto e viene riportato un errore 2085 (MQRC_UNKNOWN_OBJECT_NAME) .

IBM MQ Advanced Message Security (AMS) viene abilitato automaticamente in un client IBM MQ e quindi, per impostazione predefinita, il client tenta di controllare le politiche di sicurezza per gli oggetti nel gestore code.

I seguenti esempi mostrano come utilizzare l'attributo DisableClientAMS :

- Per disabilitare AMS:

```
Security:  
DisableClientAMS=Yes
```

- Per abilitare AMS:

```
Security:  
DisableClientAMS=No
```

Questo attributo può essere letto da client C, IBM MQ classes for Java e IBM MQ classes for JMS .

MQIInitialKeyFile = nomepercorso

Il percorso completo e il nome del file contenente la chiave iniziale utilizzata per codificare le credenziali fornite dal client. La chiave iniziale deve essere specificata se è stato specificato un file di chiavi iniziale quando la passphrase del repository chiavi è stata codificata utilizzando il programma di utilità **runmqicred** .

Questo attributo può essere letto da C e da client .NET non gestiti.

Attività correlate

[Disabilitazione di Advanced Message Security sul client](#)

Stanza SSL del file di configurazione client

Utilizzare la stanza SSL per specificare le informazioni sull'utilizzo di TLS.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare [Quali attributi IBM MQ possono essere letti da ciascun client](#).

I seguenti attributi possono essere inclusi nella stanza SSL:

OutboundSNI = CANALE | NOME HOST

Se **OutboundSNI** è impostato su CHANNEL, i client con capacità SNI impostano SNI sul nome canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS.

Se questo attributo è impostato su HOSTNAME, i client che supportano SNI impostano l'intestazione SNI sul nome host, causando la ricezione da parte delle richieste di connessione in uscita del certificato predefinito del gestore code remoto durante l'handshake TLS e quindi non è possibile utilizzare i certificati per canale.

Questo attributo può essere letto da client C, .NET non gestiti, IBM MQ classes for Java e IBM MQ classes for JMS .

I valori delle proprietà vengono interpretati in modo sensibile al maiuscolo / minuscolo dal client Java/JMS , quindi i valori YES/NO devono essere impostati in maiuscolo.

Da IBM MQ 9.3.0, il client IBM MQ gestito .NET è stato aggiornato per impostare SERVERNAME sul rispettivo nome host se la proprietà **OutboundSNI** è impostata su HOSTNAME, che consente a un client IBM MQ gestito .NET di connettersi a un gestore code utilizzando [Red Hat OpenShift routes](#).

Nota: Se un'applicazione con un'impostazione **OutboundSNI** di HOSTNAME si connette a un canale con un'etichetta di certificato configurata, l'applicazione viene rifiutata con un MQRC_SSL_INITIALIZATION_ERROR e nei log degli errori del gestore code viene visualizzato un messaggio AMQ9673 .

AllowOutboundSNI = YES (predefinito) | NO

Se abilitata, i client con capacità SNI imposteranno SNI sul nome del canale IBM MQ di destinazione sul sistema remoto quando si avvia una connessione TLS. Se questo attributo è impostato su NO, i client con supporto SNI non imposteranno l'intestazione SNI causando la ricezione da parte delle richieste di connessione in uscita del certificato predefinito del gestore code remoto durante l'handshake TLS e, pertanto, i certificati per canale non possono essere utilizzati.

Questo attributo può essere letto da client C, .NET non gestiti, IBM MQ classes for Java e IBM MQ classes for JMS .

I valori delle proprietà vengono interpretati in modo sensibile al maiuscolo / minuscolo dal client Java/JMS , quindi i valori YES/NO devono essere impostati in maiuscolo.



Attenzione: Deprecated Da IBM MQ 9.3.0 la proprietà **AllowOutboundSNI** è obsoleta ed è disponibile solo per scopi di compatibilità con le versioni precedenti.

AllowOutboundSNI impostato su YES fornisce la stessa funzione di **OutboundSNI** impostata su CHANNEL, mentre **AllowOutboundSNI** impostato su NO fornisce la stessa funzione di **OutboundSNI** impostata su HOSTNAME.

Se nella stanza SSL sono presenti sia gli attributi **AllowOutboundSNI** che **OutboundSNI** , l'impostazione di **OutboundSNI** ha la precedenza.

IBM I ALW AllowTLSV13 =Y | YES | T | TRUE (valore predefinito) | N | NO | F |

FALSE

Specifica se un gestore code è in grado di utilizzare TLS 1.3 CipherSpecs (consultare [Abilitazione di CipherSpecs](#)).

Questo attributo può essere letto dai client C/C + +.

Questo attributo può presentare i seguenti valori:

- Y (predefinito), YES (predefinito), T (predefinito) o TRUE (predefinito): abilita TLS 1.3 che consente al gestore code di utilizzare TLS 1.3 CipherSpecs.
- N, NO, F o FALSE: disabilita TLS 1.3, il che significa che il gestore code non è in grado di utilizzare i CipherSpecs TLS 1.3 .

Nota: Quando si utilizza il client MQI, il valore **AllowTLSV13** viene dedotto a meno che non venga specificato esplicitamente nella sezione SSL del file “Stanza SSL del file di configurazione client” a pagina 194 utilizzato dall'applicazione. Per ulteriori informazioni, vedi [IBM MQ MQI client and TLS 1.3](#).

CDPCheckExtensions = YES|NO (predefinito)

CDPCheckExtensions specifica se i canali TLS su questo gestore code tentano di verificare i server CDP denominati nelle estensioni del certificato CrlDistributionPoint.

Questo attributo può essere letto da client C e .NET non gestiti.

Questo attributo può presentare i seguenti valori:

- YES (valore predefinito): i canali TLS tentano di controllare i server CDP per stabilire se un certificato digitale è revocato.
- NO: i canali TLS non tentano di controllare i server CDP. Questo è il valore predefinito.

CertificateLabel = stringa

L'etichetta del certificato della definizione del canale.

Questo attributo può essere letto da client C e .NET non gestiti.

Consultare [Etichetta certificato \(CERTLABL\)](#) per ulteriori informazioni.

CertificateValPolitica = stringa

Determina il tipo di convalida del certificato utilizzato.

Questo attributo può essere letto da client C e .NET non gestiti.

Questo attributo può presentare i seguenti valori:

ANY

Utilizzare qualsiasi politica di convalida del certificato supportata dalla libreria dei socket protetti sottostante. Questa è l'impostazione predefinita.

RFC5280

Utilizzare solo la convalida del certificato conforme allo standard RFC 5280.



Non utilizzare la convalida del certificato.

ClientRevocationAssegni = REQUIRED|OPTIONAL|DISABLED

Determina il modo in cui è configurato il controllo della revoca del certificato se la chiamata di connessione client utilizza un canale TLS. Vedere anche [OCSPAuthentication](#).

Questo attributo può essere letto da client C e .NET non gestiti.

Questo attributo può presentare i seguenti valori:

OBBLIGATORIO (valore predefinito)

Tenta di caricare la configurazione della revoca del certificato da CCDT ed esegue il controllo della revoca come configurato. Se il file CCDT non può essere aperto o non è possibile convalidare il certificato (ad esempio, perché un server OCSP o CRL non è disponibile), la chiamata MQCONN ha esito negativo. Non viene eseguito alcun controllo di revoca se CCDT non contiene alcuna configurazione di revoca, ma ciò non causa l'esito negativo del canale.

Windows Sui sistemi Windows, è anche possibile utilizzare Active Directory per il controllo della revoca CRL. Non è possibile utilizzare Active Directory per il controllo della revoca OCSP.

Se si utilizza MQSCO o CCDT, la connessione ha esito positivo. Se non è presente alcun file CCDT e se MQSCO non viene fornito, la connessione non riesce con un codice di errore 2059 e il log degli errori riporta AMQ9518E: File '/var/mqm/AMQCLCHL.TAB' non trovato.

Facoltativo

Come per REQUIRED, ma se non è possibile caricare la configurazione di revoca del certificato, il canale non ha esito negativo.

DISABILITATO

Non è stato effettuato alcun tentativo di caricare la configurazione di revoca del certificato da CCDT e non è stato eseguito alcun controllo di revoca del certificato.

Nota: Se si utilizza MQCONNX invece delle chiamate MQCONN, è possibile scegliere di fornire i record delle informazioni di autenticazione (MQAIR) tramite MQSCO. Il funzionamento predefinito con MQCONNX non ha quindi esito negativo se il file CCDT non può essere aperto, ma presuppone che si stia fornendo un MQAIR (anche se si sceglie di non farlo).

EncryptionPolicySuiteB = *stringa*

Determina se un canale utilizza la crittografia conforme a Suite - B e quale livello di potenza deve essere utilizzato.

Questo attributo può essere letto da client C e .NET non gestiti.

Questo attributo può presentare i seguenti valori:

Nessuna

La crittografia compatibile Suite - B non viene utilizzata. Questa è l'impostazione predefinita.

128_BIT,192_BIT

Imposta il livello di sicurezza su entrambi i livelli a 128 bit e 192 bit.

128_BIT

Imposta il livello di sicurezza a 128 bit.

192_BIT

Imposta il livello di sicurezza su 192 bit.

ALW**EnvironmentScope=PROCESS|CONNESSIONE**

Controlla se IBM MQ utilizza un singolo ambiente IBM Global Security Kit (GSKit) per l'intero processo o un ambiente GSKit per connessione.

Questo attributo può essere letto dai client C.

Questo attributo può presentare i seguenti valori:

PROCESS

Un singolo ambiente GSKit viene utilizzato per più connessioni create dal processo. L'uso di questa impostazione indica che le modifiche del keystore TLS non saranno disponibili fino a quando tutte le connessioni TLS attive all'interno del processo non saranno arrestate.

Questo è il valore predefinito.

CONNECTION

Viene creato un ambiente GSKit per ogni connessione all'interno dello stesso processo. L'abilitazione di questa opzione indica che le modifiche al keystore TLS verranno acquisite immediatamente da qualsiasi nuova connessione TLS avviata dal processo.



Avvertenza: L'abilitazione di questa modalità operativa fa sì che le applicazioni utilizzino ulteriori risorse CPU e memoria per creare ogni ambiente GSKit. Questo consumo di risorse aumenta con ogni ulteriore connessione TLS simultanea.

ALW**MinimumRSAKeySize=int**

Specifica la dimensione chiave minima che i certificati RSA devono avere per essere accettati. Consente qualsiasi valore uguale o superiore a 0. Se non specificato, il valore predefinito è 1.

Questo attributo può essere letto dai client C/C + +.

OCSPAuthentication = FACOLTATIVO|OBBLIGATORIO|AVVERTENZA

Definisce il comportamento di IBM MQ quando OCSP è abilitato e il controllo della revoca OCSP non è in grado di determinare lo stato della revoca del certificato. Vedere anche **ClientRevocationChecks**.

Questo attributo può essere letto da client C e .NET non gestiti.

Questo attributo può presentare i seguenti valori:

Facoltativo

Qualsiasi certificato con uno stato di revoca che non può essere determinato dal controllo OCSP viene accettato e non viene generato alcun avviso o messaggio di errore. La connessione SSL o TLS continua come se non fosse stato effettuato alcun controllo di revoca.

OBBLIGATORIO

Il controllo OCSP deve produrre un risultato di revoca definitivo per ogni certificato SSL o TLS che viene controllato. Qualsiasi certificato SSL o TLS con uno stato di revoca che non può essere verificato viene rifiutato con un messaggio di errore. Se i messaggi di evento SSL del gestore code sono abilitati, viene generato un messaggio MQRQ_CHANNEL_SSL_ERROR con un ReasonQualifier di MQRQ_SSL_HANDSHAKE_ERROR. La connessione è chiusa.

Questo è il valore predefinito.

WARN

Un'avvertenza viene riportata nei log degli errori del gestore code se un controllo della revoca OCSP non è in grado di determinare lo stato di revoca di un certificato SSL o TLS. Se i messaggi di evento SSL del gestore code sono abilitati, viene generato un messaggio MQRQ_CHANNEL_SSL_WARNING con un ReasonQualifier di MQRQ_SSL_UNKNOWN_REVOCATION. La connessione può continuare.

OCSPCheckExtensions = YES|NO

Controlla se IBM MQ agisce sulle estensioni del certificato di accesso AuthorityInfo.

Questo attributo può essere letto da client C e .NET non gestiti.

Se il valore è impostato su NO, IBM MQ ignora le estensioni del certificato di accesso AuthorityInfo non tenta un controllo di sicurezza OCSP. Il valore predefinito è Sì.

ALW **OCSPTimeout = numero**

Il numero di secondi di attesa per un responder OCSP durante l'esecuzione di una verifica di revoca.

Questo attributo può essere letto da client C e .NET non gestiti.

Da IBM MQ 9.3.0, se è impostato il valore 0, viene utilizzato il timeout predefinito di 30 secondi.

Se non viene impostato alcun valore, viene utilizzato il valore predefinito IBM MQ di 30 secondi.

ALW **PeerCertChainValidation=stringa**

Questo attributo può essere letto da C e da client .NET non gestiti.

La stringa può essere uno dei due valori:

- **Usepeerchain [Valore predefinito]**: la catena di certificati fornita dal peer può essere utilizzata per colmare eventuali lacune della catena di attendibilità durante la convalida dei certificati. Con l'eccezione del certificato root.
- **Truststoreonly [Non consigliato]**: verranno utilizzati solo i certificati nel truststore per convalidare il certificato del peer.

SSLCryptoHardware = stringa

Imposta la stringa del parametro richiesta per configurare l'hardware crittografico PKCS #11 presente sul sistema.

Questo attributo può essere letto da client C e .NET non gestiti.

Specificare una stringa nel seguente formato: *GSK_PKCS11 = driver path and filename;token label;token password;symmetric cipher setting;*

Ad esempio: *GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;passwd;SYMMETRIC_CIPHER_ON*

Il percorso del driver è un percorso assoluto della libreria condivisa che fornisce il supporto per la scheda PKCS #11 . Il nome file del driver è il nome della libreria condivisa. Un esempio del valore richiesto per il percorso del driver PKCS #11 e il nome file è */usr/lib/pkcs11/PKCS11_API.so*. Per accedere alle operazioni di cifratura simmetrica tramite GSKit, specificare il parametro di impostazione della cifratura simmetrica. Il valore di questo parametro è:

SYMMETRIC_CIPHER_OFF

Non accedere alle operazioni di cifratura simmetrica. Questa è l'impostazione predefinita.

SIMMETRICA_CIFRA_ON

Accedere alle operazioni di cifratura simmetriche.

Linux

AIX

Quando si forniscono i diversi componenti della stringa, è necessario eseguire l'escape dei caratteri punto e virgola utilizzando il carattere barra retroversa, poiché il carattere punto e virgola viene considerato come un commento. Ad esempio: `'\ ;'`

È necessario proteggere la password token contenuta nella stringa di attributo

SSLCryptoHardware . Per ulteriori informazioni, consultare [IBM MQ clients using cryptographic hardware](#) .

Per gestire le parole d'ordine codificate, non esiste ora alcun limite alla lunghezza della stringa.

Il valore predefinito è uno spazio vuoto. Se si specifica una stringa non nel formato corretto, viene generato un errore.

SSLCryptoHardwareKeyFile = nomepercorso

Il percorso completo e il nome del file contenente la chiave iniziale utilizzata per codificare la password nella stringa di configurazione hardware di crittografia PKCS #11 specificata con l'attributo **SSLCryptoHardware** . La chiave iniziale deve essere specificata se è stato specificato un file di chiavi iniziale quando la parola d'ordine nella stringa di configurazione dell'hardware di crittografia è

stata codificata utilizzando il comando **runp11cred** . Per ulteriori informazioni, consultare [ClientIBM MQ](#) che utilizzano hardware crittografico.

Questo attributo può essere letto da C e da client .NET non gestiti.

SSLFipsRequired = YES|NO

Specifica se devono essere utilizzati solo algoritmi certificati FIPS se la crittografia viene eseguita in IBM MQ.

Questo attributo può essere letto da C e da client .NET non gestiti.

Se l'hardware di crittografia è configurato, i moduli di crittografia utilizzati sono quei moduli forniti dal prodotto hardware. Questi potrebbero, o meno, essere certificati FIPS ad un particolare livello, a seconda del prodotto hardware in uso.

SSLHTTPProxyName = stringa

La stringa è il nome host o l'indirizzo di rete del server proxy HTTP che deve essere utilizzato da GSKit per i controlli OCSP. Questo indirizzo può essere seguito da un numero di porta facoltativo, racchiuso tra parentesi. Se non si specifica alcun numero, viene utilizzata la porta HTTP predefinita (80).

Questo attributo può essere letto da C e da client .NET non gestiti.

AIX Per a 32-bit su AIX, l'indirizzo di rete può essere solo un indirizzo IPv4 .

Su altre piattaforme, l'indirizzo di rete può essere un indirizzo IPv4 o IPv6 .

Questo attributo potrebbe essere necessario se, ad esempio, un firewall impedisce l'accesso all'URL del responder OCSP.

ALW SSLHTTPConnectTimeout = numero|0

Il numero di secondi di attesa per stabilire correttamente una connessione di rete a un server HTTP durante l'esecuzione di un controllo di revoca.

Questo attributo può essere letto da client C e .NET non gestiti.

Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ 0 (off).

SSLKeyRepository = nomepercorso

Il percorso completo e il nome file del repository chiavi che contiene il certificato digitale dell'utente. Se l'estensione file non è specificata, si presume che sia .kdb.

Questo attributo può essere letto da C e da client .NET non gestiti.

SSLKeyRepositoryPassword = passphrase

La passphrase per accedere al repository delle chiavi. Il valore può essere una stringa di testo semplice o una passphrase che è stata crittografata utilizzando il programma di utilità **runmqicred** .

Questo attributo può essere letto da C e da client .NET non gestiti.

SSLKeyResetConteggio = intero|0

Il numero di byte non codificati inviati e ricevuti su un canale TLS prima che la chiave segreta venga rinegoziata.

Questo attributo può essere letto da C e da client .NET non gestiti.

Il valore deve essere compreso tra 0 e 999999999.

Il valore predefinito è 0, che significa che le chiavi segrete non vengono mai rinegoziate.

Se si specifica un valore compreso tra 1 e 32768, i canali TLS utilizzano un conteggio di reimpostazione della chiave segreta pari a 32768 (32Kb). Ciò per evitare un numero eccessivo di reimpostazioni della chiave, che si verificherebbe per i valori di reimpostazione della chiave segreta.

Multi Stanza TCP del file di configurazione client

Utilizzare la stanza TCP per specificare i parametri di configurazione del protocollo di rete TCP.

Nota: La descrizione di ciascun attributo di questa stanza indica quali client IBM MQ possono leggere tale attributo. Per una tabella di riepilogo per tutte le stanze del file di configurazione IBM MQ MQI client, consultare Quali attributi IBM MQ possono essere letti da ciascun client.

I seguenti attributi possono essere inclusi nella stanza TCP:

ClntRcvBuffSize = numero|0

La dimensione, in byte, del buffer di ricezione TCP/IP utilizzato dall'estremità client di un canale di connessione server di connessione client.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

Un valore zero indica che il sistema operativo gestirà le dimensioni del buffer, in contrapposizione alle dimensioni del buffer fissate da IBM MQ. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

ClntSndBuffSize = numero|0

La dimensione in byte del buffer di invio TCP/IP utilizzato dall'estremità client di un canale di connessione server di connessione client.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

Un valore zero indica che il sistema operativo gestirà le dimensioni del buffer, in contrapposizione alle dimensioni del buffer fissate da IBM MQ. Se il valore è impostato su 0, vengono utilizzati i valori predefiniti del sistema operativo. Se non viene impostato alcun valore, viene utilizzato il valore predefinito di IBM MQ , 32768.

Connect_Timeout = numero

Il numero di secondi prima del timeout di un tentativo di connessione del socket.

Se **ConnectTimeout** = 0 e SOCK_NONBLOCK vengono emessi prima di una chiamata connect () asincrona, la chiamata non è bloccata. Il valore di timeout predefinito di 20 secondi (CONNECT_WAIT_MAX) è applicabile per il controllo dello stato di connessione.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Javae IBM MQ classes for JMS .

I processi del canale IBM MQ si collegano su socket non bloccanti. Pertanto, se l'altra estremità del socket non è pronta, connect () restituisce immediatamente *EINPROGRESS* o *EWOULDBLOCK*. In seguito, non è stato effettuato alcun tentativo di riconnessione.

Se Connect_Timeout è impostato su un valore diverso da zero, IBM MQ attende il periodo stabilito sulla chiamata select () affinché il socket sia pronto. Ciò aumenta le possibilità di riuscita di una chiamata connect () successiva. Questa opzione potrebbe essere utile in situazioni in cui le connessioni richiedono un periodo di attesa, a causa di un carico elevato sulla rete.

Non esiste alcuna relazione tra i parametri Connect_Timeout, ClntSndBuffSize e ClntRcvBuffSize .

IPAddressVersion = MQIPADDR_IPV4|MQIPADDR_IPV6

Specifica quale protocollo IP utilizzare per una connessione di canale.

Questo attributo può essere letto da client C, .NETnon gestiti, .NETgestiti e XMS .NET gestiti.

Ha i valori di stringa possibili di MQIPADDR_IPV4 o MQIPADDR_IPV6. Questi valori hanno lo stesso significato di IPV4 e IPV6 in **ALTER QMGR IPADDRV** e la variabile di ambiente **MQIPADDRV** .

KeepAlive = S|NO

Attivare o disattivare la funzione KeepAlive . KeepAlive=YES fa sì che TCP/IP controlli periodicamente che l'altra estremità della connessione sia ancora disponibile. In caso contrario, il canale viene chiuso.

Questo attributo può essere letto da client C, .NETnon gestiti, IBM MQ classes for Java, IBM MQ classes for JMS, .NETgestiti e XMS .NET gestiti.

Windows **Library1 = NomeDLL|WSOCK32**

(Solo Windows) Il nome della DLL socket TCP/IP.

Questo attributo può essere letto da client C e .NET non gestiti.

V 9.4.0 **Stanza di traccia del file di configurazione client**

Utilizzare la stanza Trace per abilitare la traccia per le librerie client IBM MQ .NET e XMS .NET .

I seguenti attributi possono essere inclusi nella stanza TRACE:

MQDotnetTraceLevel=0 (predefinito) |1|2

L'attributo **MQDotnetTraceLevel1** viene utilizzato per avviare o arrestare la traccia IBM MQ .NET :

- 0: arresta la traccia - questo è il valore predefinito.
- 1: avvia la traccia con meno dettagli.
- 2: Avvia la traccia con dettagli completi - consigliato.

Questo attributo può essere letto dal client IBM MQ .NET gestito.

MQDotnetTracePercorso =nomepercorso

L'attributo **MQDotnetTracePath** fa riferimento a una cartella in cui verranno creati i file di traccia di IBM MQ .NET . La directory corrente dell'applicazione viene utilizzata se il percorso è vuoto o se la proprietà **MQDotnetTracePath** non è definita.

Questo attributo può essere letto dal client IBM MQ .NET gestito.

MQDotnetErrorPercorso =nomepercorso

L'attributo **MQDotnetErrorPath** fa riferimento a una cartella in cui verranno creati i file di log degli errori per la traccia IBM MQ .NET . La directory corrente dell'applicazione viene utilizzata se il percorso è vuoto o se l'attributo **MQDotnetErrorPath** non è definito.

Questo attributo può essere letto dal client IBM MQ .NET gestito.

XMSDotnetTraceLevel=0 (predefinito) |1|2

L'attributo **XMSDotnetTraceLevel1** viene utilizzato per avviare o arrestare la traccia XMS .NET :

- 0: arresta la traccia - questo è il valore predefinito.
- 1: avvia la traccia con il formato di base.
- 2: avvia la traccia con il formato avanzato.

Questo attributo può essere letto dal client XMS .NET gestito.

XMSDotnetTraceFilePath=nome file

Se un valore non è impostato per l'attributo **XMSDotnetTraceFilePath** o se questo attributo è presente ma contiene una stringa vuota, il file di traccia per XMS .NET viene posizionato nella directory corrente. Per memorizzare il file di traccia in una directory denominata, specificare il nome della directory in **XMSDotnetTraceFilePath**, ad esempio **XMSDotnetTraceFilePath="c:\somepath"**.

Questo attributo può essere letto dal client XMS .NET gestito.

XMSDotnetTraceSpecifica =ComponentName=tipo=stato

L'attributo **XMSDotnetTraceSpecification** specifica il nome della classe che si desidera tracciare e il tipo di traccia richiesto per XMS .NET:

- *ComponentName* è il nome della classe che si desidera tracciare. È possibile utilizzare un carattere jolly * in questo nome. Ad esempio, *=all=enabled specifica che si desidera tracciare tutte le classi e IBM.XMS.impl.*=all=enabled specifica che si richiede solo la traccia API.
- *type* può essere uno dei seguenti tipi di traccia: all, debug, event, EntryExit.
- *state* può essere abilitato o disabilitato.

È possibile raggruppare più elementi di traccia utilizzando un delimitatore ':' (due punti).

Questo attributo può essere letto dal client XMS .NET gestito.

XMSDotnetTraceFileSize=dimensione

L'attributo **XMSDotnetTraceFileSize** specifica la dimensione massima del file di traccia che deve essere generato per XMS .NET. Il valore massimo predefinito è 20 MB, specificato come XMSDotnetTraceFileSize=20.

Questo attributo può essere letto dal client XMS .NET gestito.

XMSDotnetTraceFileNumber=numero

L'attributo **XMSDotnetTraceFileNumber** specifica il numero di file di traccia da conservare per XMS .NET. Il valore predefinito è 4 (un file attivo e tre file di archiviazione). Il numero minimo consentito è 2.

Questo attributo può essere letto dal client XMS .NET gestito.

Attività correlate

[Traccia delle applicazioni IBM MQ .NET con mqclient.ini](#)

[Traccia delle applicazioni XMS .NET con mqclient.ini](#)

Multi

File di configurazione della traccia di attività, mqat.ini

Il file di configurazione della traccia attività, mqat . ini, viene utilizzato per configurare il comportamento della traccia attività. Questo file viene utilizzato per definire il livello e la frequenza dei dati di traccia delle attività di report. Il file fornisce anche un modo per definire le regole per abilitare e disabilitare la traccia dell'attività in base al nome di una applicazione.

Il file mqat . ini segue lo stesso formato della coppia chiave stanza e parametro - valore dei file mqs . ini e qm . ini . Il file è composto da una singola stanza, AllActivityTrace, utilizzata per configurare il livello e la frequenza dei dati di traccia delle attività di report per impostazione predefinita per tutta la traccia delle attività. Il file può contenere anche più stanze ApplicationTrace . Ciascuna di queste stanze definisce una regola per il funzionamento della traccia per una o più connessioni, in base alla corrispondenza del nome dell'applicazione delle connessioni alla regola. Per ulteriori informazioni, consultare [Application activity trace](#) e [Configuring activity trace behavior using mqat . ini](#).

Il gestore code applica una serie di regole per determinare quali impostazioni di stanza utilizzare per una connessione. Facoltativamente, è possibile sovrascrivere il livello di traccia globale e le impostazioni di frequenza nella stanza AllActivityTrace per le connessioni che corrispondono a una stanza ApplicationTrace . Per ulteriori informazioni, vedere [Configurazione del comportamento della traccia dell'attività utilizzando mqat . ini](#).

Percorsi di directory

IBM i **Linux** **AIX** Sui sistemi AIX and Linux e IBM i, mqat . ini si trova nella directory dei dati del gestore code, che è la stessa ubicazione del file qm . ini .

Windows Sui sistemi Windows , mqat . ini si trova nella directory dei dati del gestore code C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name. Gli utenti che eseguono le applicazioni da tracciare hanno bisogno dell'autorizzazione per leggere questo file.

Multi

AllActivityStanza di traccia del file mqat.ini

La stanza di traccia AllActivitydel file di configurazione mqat . ini specifica i parametri utilizzati per configurare i livelli di traccia per un gestore code.

Una singola stanza di traccia AllActivitydefinisce le impostazioni per la traccia dell'attività che viene applicata a tutte le connessioni IBM MQ , a meno che non venga sovrascritta.

I singoli valori nella stanza AllActivityTrace possono essere sovrascritti da informazioni più specifiche in una stanza [ApplicationTrace](#).

Se viene specificata più di una stanza di traccia AllActivity, vengono utilizzati i valori nell'ultima stanza. I parametri mancanti dalla traccia AllActivityselezionata assumono valori predefiniti. I parametri e i valori delle stanze di traccia AllActivityprecedenti vengono ignorati.

ActivityInterval

L'intervallo di tempo in secondi tra i messaggi di traccia. La traccia attività non utilizza un thread timer, quindi il messaggio di traccia non viene scritto nel momento esatto in cui il tempo trascorre, ma viene scritto quando viene eseguita la prima operazione MQI una volta trascorso l'intervallo di tempo. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando viene raggiunto il conteggio attività). L'impostazione predefinita è 1.

ActivityCount

Il numero di operazioni MQI tra i messaggi di traccia. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando trascorre l'intervallo di attività). Il valore predefinito è 100.

TraceLevel

La quantità di dettagli del parametro tracciata per ogni operazione. La descrizione delle singole operazioni descrive i parametri inclusi per ciascun livello di traccia. Impostare su LOW, MEDIUM o HIGH. Il valore predefinito è MEDIUM.

Dati TraceMessage

La quantità di dati del messaggio tracciati in byte per le operazioni MQGET, MQPUT, MQPUT1e Callback. Il valore predefinito è 0.

Messaggio StopOnGetTrace

Può essere impostato su ON o OFF. L'impostazione predefinita è ON.

SubscriptionDelivery

Può essere impostato su BATCH o IMMEDIATE. Determina se i parametri **ActivityInterval** e **ActivityCount** devono essere utilizzati quando sono presenti una o più sottoscrizioni di traccia dell'attività. L'impostazione di questo parametro su IMMEDIATE determina la sovrascrittura dei valori **ActivityInterval** e **ActivityCount** con i valori effettivi di 1 quando i dati di traccia hanno una sottoscrizione corrispondente. Ogni record di traccia dell'attività non viene raggruppato in batch con altri record della stessa connessione e viene invece consegnato alla sottoscrizione immediatamente senza alcun ritardo. L'impostazione IMMEDIATE aumenta il sovraccarico delle prestazioni della raccolta dei dati di traccia dell'attività. L'impostazione predefinita è BATCHED.

Attività correlate

[Configurazione del comportamento della traccia attività utilizzando mqat.ini](#)

Multi Stanza ApplicationTrace del file mqat.ini

Il file di configurazione mqat . ini può contenere più stanze ApplicationTrace . Ognuna di queste stanze definisce una regola per il comportamento della traccia per una o più connessioni, in base alla corrispondenza del nome dell'applicazione delle connessioni alla regola.

È possibile impostare i seguenti valori per la stanza ApplicationTrace :

Traccia

Interruttore di traccia dell'attività che può essere impostato su ON o OFF. Il parametro **Trace** è un parametro obbligatorio senza valore predefinito. Può essere utilizzato nella sezione specifica dell'applicazione per stabilire se la traccia dell'attività è attiva per l'ambito della stanza dell'applicazione corrente. Notare che questo valore sovrascrive le impostazioni **ACTVTRC** e **ACTVCONO** per il gestore code.

App1Name

Il parametro **App1Name** è specificato come stringa di caratteri ed è un parametro obbligatorio senza valore predefinito. Questo valore viene utilizzato per stabilire a quali applicazioni si applica la sezione ApplicationTrace . Viene associato al valore **App1Name** dalla struttura del contesto dell'uscita API (che è equivalente a MQMD.PutApp1Name). Il contenuto del valore **App1Name** varia a seconda dell'ambiente dell'applicazione.

Su Multiplatforms, solo la parte del nome file di MQAXC MQAXC.ApplName corrisponde al valore nella sezione. I caratteri a sinistra del separatore di percorso più a destra vengono ignorati quando viene effettuato il confronto.

Un singolo carattere jolly (*) può essere utilizzato alla fine del valore **App1Name** per corrispondere a qualsiasi numero di caratteri dopo tale punto. Se il valore **App1Name** è impostato su un singolo carattere jolly (*), il valore **App1Name** corrisponde a tutte le applicazioni.

IBM i **App1Function**

Il parametro **App1Function** viene specificato come stringa di caratteri. Il valore predefinito è *. Il valore di questo parametro viene utilizzato per qualificare i programmi applicativi a cui si applicano la stanza ApplicationTrace e il valore **App1Name**.

La stanza è facoltativa ed è valida solo per i gestori code IBM i. Un singolo carattere jolly (*) può essere utilizzato alla fine del valore **App1Name** per corrispondere a qualsiasi numero di caratteri. Ad esempio, una stanza ApplicationTrace che specifica **App1Name** = * e **App1Function** = AMQSPUTO si applica a tutti i richiami del programma AMQSPUTO da qualsiasi lavoro.

App1Class

Il parametro **App1Class** definisce la classe di una applicazione e può essere impostata sui seguenti valori:

- UTENTE
- MCA
- TUTTO (questo è il valore predefinito)

Per una spiegazione del modo in cui i valori **AppType** corrispondono alle connessioni IBM MQ, consultare la [Tabella 3 in Configurazione del comportamento della traccia dell'attività utilizzando mqat.ini](#).

Facoltativamente, il livello di traccia globale e le impostazioni di frequenza nella stanza di traccia AllActivity possono essere sovrascritti per quelle connessioni che corrispondono a una stanza ApplicationTrace.

I parametri seguenti possono essere impostati in una stanza ApplicationTrace. Se non sono impostati, il valore viene ereditato dalle impostazioni della [stanza di tracciaAllActivity](#):

ActivityInterval

L'intervallo di tempo in secondi tra i messaggi di traccia. La traccia attività non utilizza un thread timer, quindi il messaggio di traccia non viene scritto nel momento esatto in cui il tempo trascorre, ma viene scritto quando viene eseguita la prima operazione MQI una volta trascorso l'intervallo di tempo. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando viene raggiunto il conteggio attività). L'impostazione predefinita è 1.

ActivityCount

Il numero di operazioni MQI tra i messaggi di traccia. Se questo valore è 0, il messaggio di traccia viene scritto quando la connessione si disconnette (o quando trascorre l'intervallo di attività). Il valore predefinito è 100.

TraceLevel

La quantità di dettagli del parametro tracciata per ogni operazione. La descrizione delle singole operazioni descrive i parametri inclusi per ciascun livello di traccia. Impostare su LOW, MEDIUM o HIGH. Il valore predefinito è MEDIUM.

Dati TraceMessage

La quantità di dati del messaggio tracciati in byte per le operazioni MQGET, MQPUT, MQPUT1e Callback. Il valore predefinito è 0.

Messaggio StopOnGetTrace

Può essere impostato su ON o OFF. L'impostazione predefinita è ON.

Attività correlate

[Configurazione del comportamento della traccia attività utilizzando mqat.ini](#)

Configurazione dell'accodamento distribuito



Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ, incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

Prima di iniziare

Prima di leggere questa sezione è utile avere una conoscenza dei canali, delle code e degli altri concetti introdotti in [Accodamento distribuito e cluster](#).

Se è necessario connettere due gestori code che si trovano su reti fisiche differenti o è necessario comunicare tramite firewall, l'utilizzo di IBM MQ Internet Pass-Thru potrebbe semplificare la configurazione. Per ulteriori informazioni, consultare [IBM MQ Internet Pass-Thru](#).

Procedura

- Utilizzare le informazioni contenute nei seguenti argomenti secondari per collegare le applicazioni utilizzando l'accodamento distribuito:
 - [“Tecniche di accodamento distribuito IBM MQ”](#) a pagina 206
 - [“Introduzione alla gestione delle code distribuite”](#) a pagina 226
 - [“Come inviare un messaggio a un altro gestore code”](#) a pagina 229
 - [“Attivazione dei canali”](#) a pagina 250
 - [“Sicurezza dei messaggi”](#) a pagina 248
 -  [“Monitoraggio e controllo dei canali su AIX, Linux, and Windows”](#) a pagina 258
 -  [“Monitoraggio e controllo dei canali su IBM i”](#) a pagina 282

Concetti correlati

[“Setting up IBM MQ for z/OS”](#) a pagina 888

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

Attività correlate

[“Configurazione delle connessioni tra client e server”](#) a pagina 16

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

[“Configurazione di un cluster di gestore code”](#) a pagina 303

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. È possibile definire componenti cluster e creare e gestire cluster.

[“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms”](#) a pagina 93

È possibile modificare il comportamento di IBM MQ o di un singolo gestore code per adattarlo alle esigenze della propria installazione modificando le informazioni nei file di configurazione (.ini). È anche possibile modificare le opzioni di configurazione per IBM MQ MQI clients.

[“Configuring queue managers on z/OS”](#) a pagina 883

Use these instructions to configure queue managers on IBM MQ for z/OS.





[“Setting up communications with other queue managers on z/OS”](#) a pagina 961

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

Tecniche di accodamento distribuito IBM MQ

Gli argomenti secondari in questa sezione descrivono le tecniche che sono di uso durante la pianificazione dei canali. Questi argomenti secondari descrivono le tecniche per pianificare come collegare i gestori code e gestire il flusso di messaggi tra le applicazioni.

Per esempi di pianificazione del canale di messaggi, consultare:

-  [Esempio di pianificazione del canale dei messaggi per AIX, Linux, and Windows](#)
-  [Esempio di pianificazione del canale dei messaggi per IBM i](#)
-  [Esempio di pianificazione del canale dei messaggi per z/OS](#)
-  [Esempio di pianificazione del canale dei messaggi per z/OS utilizzo di gruppi di condivisione code](#)

Concetti correlati

[Canali](#)

[Introduzione all'accodamento dei messaggi](#)

[Accodamento distribuito e cluster](#)

Attività correlate

[“Configurazione dell'accodamento distribuito” a pagina 205](#)

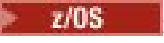
Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

Riferimenti correlati

[Informazioni di configurazione di esempio](#)

controllo del flusso di messaggi

Il controllo del flusso di messaggi è un'attività che implica l'impostazione e la manutenzione delle serie di messaggi tra gestori code. È importante per gli instradamenti che attraversano più gestori code. Questa sezione descrive come utilizzare code, definizioni di code alias e canali di messaggi sul sistema per ottenere il controllo del flusso di messaggi.

Il flusso di messaggi viene controllato utilizzando una serie di tecniche introdotte in [“Configurazione dell'accodamento distribuito” a pagina 205](#). Se il gestore code si trova in un cluster, il flusso di messaggi viene controllato utilizzando tecniche differenti, come descritto in [“controllo del flusso di messaggi” a pagina 206](#).  Se i gestori code si trovano in un gruppo di condivisione code e IGQ (intra - group queuing) è abilitato, il flusso di messaggi può essere controllato dagli agent IGQ. Questi agent sono descritti in [Accodamento all'interno del gruppo](#).

È possibile utilizzare i seguenti oggetti per ottenere il controllo del flusso di messaggi:

- Code di trasmissione
- Canali dei messaggi
- Definizione di coda remota
- Definizione alias del gestore code
- Definizione alias coda di risposta

Il gestore code e gli oggetti coda sono descritti in [Tipi di oggetti](#). I canali di messaggi sono descritti in [Componenti di accodamento distribuiti](#). Le tecniche seguenti utilizzano questi oggetti per creare flussi di messaggi nel sistema:

- Inserimento di messaggi nelle code remote
- Instradamento mediante particolari code di trasmissione
- ricezione di messaggi

- Passaggio di messaggi attraverso il sistema
- Separazione dei flussi di messaggi
- Passaggio di un flusso di messaggi a un'altra destinazione
- Risoluzione del nome della coda di risposta in un nome alias

Nota

Tutti i concetti descritti in questa sezione sono rilevanti per tutti i nodi in una rete e includono le estremità di invio e ricezione dei canali di messaggi. Per questo motivo, nella maggior parte degli esempi viene illustrato un solo nodo. L'eccezione si verifica quando l'esempio richiede una cooperazione esplicita da parte dell'amministratore all'altra estremità di un canale di messaggi.

Prima di procedere con le tecniche individuali, è utile ricapitolare i concetti di risoluzione dei nomi e i tre modi di utilizzare le definizioni di coda remota. Vedere [accodamento distribuito e cluster](#).

Concetti correlati

“Nomi coda nell'intestazione di trasmissione” a pagina 207

I nomi delle code di destinazione viaggiano con il messaggio nell'intestazione di trasmissione fino a quando non viene raggiunta la coda di destinazione.

“Come creare il gestore code e gli alias di risposta” a pagina 207

Questo argomento illustra i tre modi in cui è possibile creare una definizione di coda remota.

Nomi coda nell'intestazione di trasmissione

I nomi delle code di destinazione viaggiano con il messaggio nell'intestazione di trasmissione fino a quando non viene raggiunta la coda di destinazione.

Il nome della coda utilizzato dall'applicazione, il nome della coda logica, viene risolto dal gestore code nel nome coda di destinazione. In altre parole, il nome della coda fisica. Questo nome della coda di destinazione viaggia con il messaggio in un'area dati separata, l'intestazione di trasmissione, fino a quando non viene raggiunta la coda di destinazione. L'intestazione di trasmissione viene quindi svuotata.

Modificare la parte del gestore code di questo nome coda quando si creano classi parallele di servizio. Ricordarsi di restituire il nome del gestore code al nome originale quando è stata raggiunta la fine della deviazione della classe di servizio.

Come creare il gestore code e gli alias di risposta

Questo argomento illustra i tre modi in cui è possibile creare una definizione di coda remota.

L'oggetto di definizione della coda remota viene utilizzato in tre modi diversi. [Tabella 17 a pagina 208](#) spiega come definire ciascuno dei tre modi:

- Utilizzo di una definizione di coda remota per ridefinire un nome di coda locale.

L'applicazione fornisce solo il nome coda quando si apre una coda e questo nome coda è il nome della definizione della coda remota.

La definizione della coda remota contiene i nomi della coda di destinazione e del gestore code.

Facoltativamente, la definizione può contenere il nome della coda di trasmissione da utilizzare. Se non viene fornito alcun nome della coda di trasmissione, il gestore code utilizza il nome del gestore code, ricavato dalla definizione della coda remota, per il nome della coda di trasmissione. Se una coda di trasmissione con questo nome non è definita, ma è definita una coda di trasmissione predefinita, viene utilizzata la coda di trasmissione predefinita.

- Utilizzo di una definizione di coda remota per ridefinire un nome gestore code.

L'applicazione o il programma del canale, fornisce un nome coda insieme al nome del gestore code remoto quando si apre la coda.

Se è stata fornita una definizione di coda remota con lo stesso nome del gestore code e il nome della coda è stato lasciato vuoto, il gestore code sostituisce il nome del gestore code nella chiamata aperta con il nome del gestore code nella definizione.

Inoltre, la definizione può contenere il nome della coda di trasmissione da utilizzare. Se non viene fornito alcun nome della coda di trasmissione, il gestore code prende il nome del gestore code, preso dalla definizione della coda remota, per il nome della coda di trasmissione. Se una coda di trasmissione con questo nome non è definita, ma è definita una coda di trasmissione predefinita, viene utilizzata la coda di trasmissione predefinita.

- Utilizzo di una definizione di coda remota per ridefinire un nome di coda di risposta.

Ogni volta che un'applicazione inserisce un messaggio in una coda, può fornire il nome di una coda di risposta per i messaggi di risposta, ma con il nome del gestore code vuoto.

Se si fornisce una definizione di coda remota con lo stesso nome della coda di risposta, il gestore code locale sostituisce il nome della coda di risposta con il nome della coda dalla propria definizione.

È possibile fornire un nome gestore code nella definizione, ma non un nome coda di trasmissione.

<i>Tabella 17. Tre modi di utilizzare l'oggetto di definizione della coda remota</i>			
Utilizzo	Nome del gestore code	Nome coda	Nome coda di trasmissione
1. Definizione coda remota (su chiamata OPEN)			
Fornito nella chiamata	QM locale o vuoto	(*) richiesto	non applicabile
Fornito nella definizione	obbligatorio	obbligatorio	facoltativo
2. Alias gestore code (su chiamata OPEN)			
Fornito nella chiamata	(*) richiesto e non QM locale	obbligatorio	non applicabile
Fornito nella definizione	obbligatorio	vuoto	facoltativo
3. Alias coda di risposta (su chiamata PUT)			
Fornito nella chiamata	vuoto	(*) richiesto	non applicabile
Fornito nella definizione	facoltativo	facoltativo	vuoto

Nota: (*) indica che questo nome è il nome dell'oggetto definizione

Per una descrizione formale, consultare [Risoluzione nome coda](#).

Inserimento di messaggi nelle code remote

È possibile utilizzare oggetti di definizione della coda remota per risolvere un nome coda in una coda di trasmissione in un gestore code adiacente.

In un ambiente di accodamento distribuito, una coda di trasmissione e un canale sono il punto focale per tutti i messaggi in un'ubicazione, indipendentemente dal fatto che i messaggi provengano da applicazioni nel sistema locale o da canali provenienti da un sistema adiacente. La [Figura 6 a pagina 209](#) mostra un'applicazione che inserisce i messaggi in una coda logica denominata 'QA_norm'. La risoluzione dei nomi utilizza la definizione della coda remota 'QA_norm' per selezionare la coda di trasmissione QMB. Aggiunge quindi un'intestazione di trasmissione ai messaggi che indicano 'QA_norm at QMB'.

I messaggi provenienti dal sistema adiacente su 'Channel_back' hanno un'intestazione di trasmissione con il nome della coda fisica 'QA_norm at QMB', ad esempio. Questi messaggi non vengono modificati nella coda di trasmissione QMB.

Il canale sposta i messaggi in un gestore code adiacente.

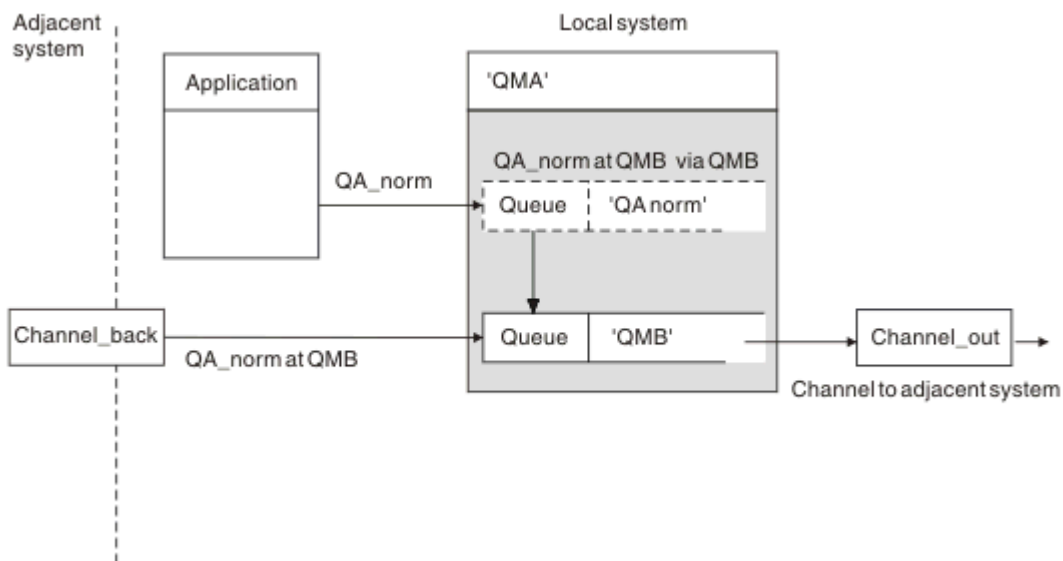


Figura 6. Una definizione di coda remota viene utilizzata per risolvere un nome coda in una coda di trasmissione in un gestore code adiacente

Se si è l'amministratore del sistema IBM MQ , è necessario:

- Definire il canale di messaggi dal sistema adiacente
- Definire il canale dei messaggi per il sistema adiacente
- Crea la coda di trasmissione QMB
- Definire l'oggetto coda remota 'QA_norm' per risolvere il nome della coda utilizzato dalle applicazioni nel nome della coda di destinazione, nel nome del gestore code di destinazione e nel nome della coda di trasmissione

In un ambiente cluster, è necessario definire solo un canale ricevente del cluster sul gestore code locale. Non è necessario definire una coda di trasmissione o un oggetto coda remota. Vedere [Cluster](#).

Ulteriori informazioni sulla risoluzione dei nomi

L'effetto della definizione della coda remota è definire un nome coda di destinazione fisica e un nome gestore code. Questi nomi vengono inseriti nelle intestazioni di trasmissione dei messaggi.

Per i messaggi in entrata da un sistema adiacente è già stato eseguito questo tipo di risoluzione dei nomi dal gestore code originale. Pertanto, hanno l'intestazione di trasmissione che visualizza il nome della coda di destinazione fisica e il nome del gestore code. Questi messaggi non sono influenzati dalle definizioni della coda remota.

Riferimenti correlati

[Risoluzione nome coda](#)

Scelta della coda di trasmissione

È possibile utilizzare una definizione di coda remota per consentire a una diversa coda di trasmissione di inviare messaggi allo stesso gestore code adiacente.

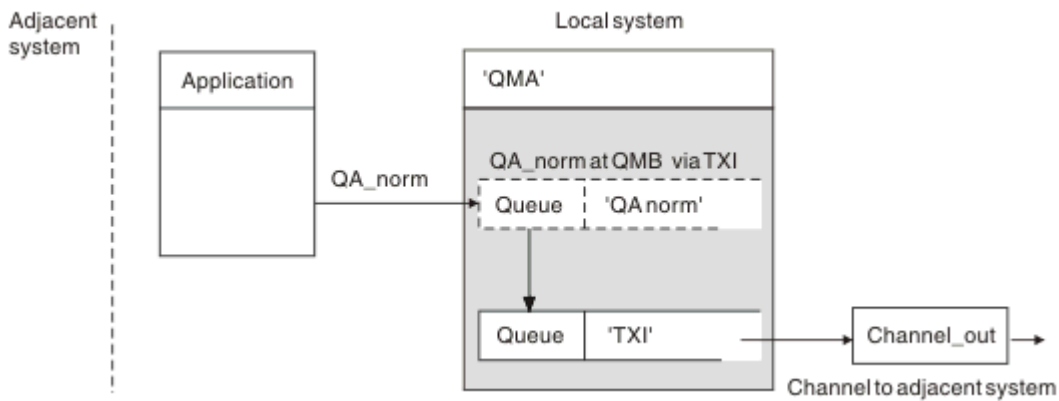


Figura 7. La definizione della coda remota consente l'utilizzo di una coda di trasmissione diversa

In un ambiente di accodamento distribuito, quando è necessario modificare un flusso di messaggi da un canale all'altro, utilizzare la stessa configurazione di sistema come mostrato in Figura 6 a pagina 209 in “Inserimento di messaggi nelle code remote” a pagina 208. Figura 7 a pagina 210 in questo argomento mostra come utilizzare la definizione della coda remota per inviare messaggi su una coda di trasmissione differente, e quindi su un canale differente, allo stesso gestore code adiacente.

Per la configurazione mostrata in Figura 7 a pagina 210, è necessario fornire l'oggetto coda remota 'QA_norm' e la coda di trasmissione 'TX1'. È necessario fornire 'QA_norm' per scegliere la coda 'QA_norm' sul Gestore code remoto, la coda di trasmissione TX1 e il gestore code 'QMB_priority'. Specificare 'TX1' nella definizione del canale adiacente al sistema.

I messaggi vengono inseriti nella coda di trasmissione 'TX1' con un'intestazione di trasmissione contenente 'QA_norm at QMB_priority' e inviati sul canale al sistema adiacente.

Il channel_back è stato lasciato fuori da questa illustrazione perché avrebbe bisogno di un alias del gestore code.

In un ambiente cluster, non è necessario definire una coda di trasmissione o una definizione di coda remota. Per ulteriori informazioni, consultare “Definizione di code cluster” a pagina 304.

ricezione di messaggi

È possibile configurare il gestore code in modo che riceva messaggi da altri gestori code. È necessario assicurarsi che non si verifichi una risoluzione del nome non intenzionale.

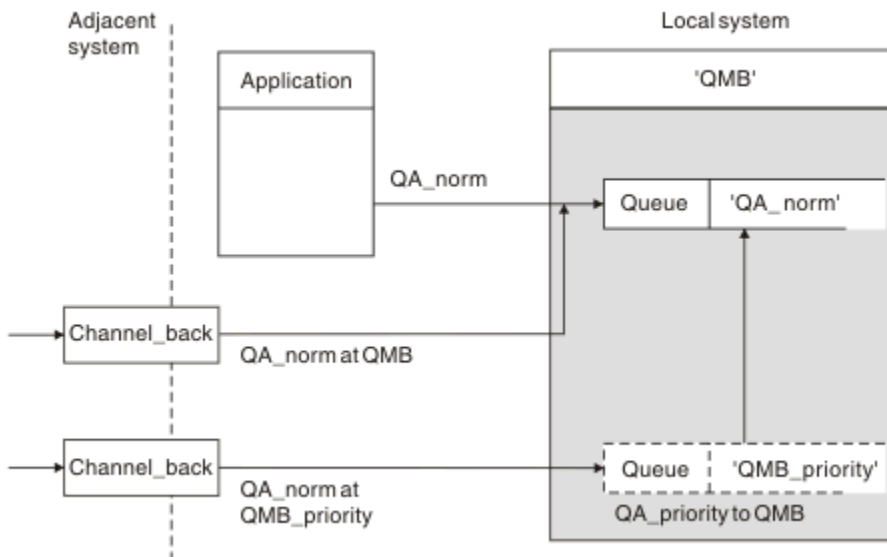


Figura 8. Ricezione diretta dei messaggi e risoluzione del nome del gestore code alias

Oltre a disporre l'invio dei messaggi, l'amministratore di sistema deve anche disporre la ricezione dei messaggi dai gestori code adiacenti. I messaggi ricevuti contengono il nome fisico del gestore code di destinazione e la coda nell'intestazione di trasmissione. Vengono trattati come messaggi da un'applicazione locale che specifica sia il nome del gestore code che il nome della coda. A causa di questo trattamento, è necessario assicurarsi che i messaggi che entrano nel sistema non abbiano una risoluzione del nome non intenzionale eseguita. Consultare [Figura 8 a pagina 210](#) per questo scenario.

Per questa configurazione, è necessario preparare:

- Canali di messaggi per ricevere messaggi da gestori code adiacenti
- Una definizione alias del gestore code per risolvere un flusso di messaggi in entrata, 'QMB_priority', nel nome del gestore code locale, 'QMB'
- La coda locale, 'QA_norm', se non esiste

Ricezione di nomi di gestori code alias

L'utilizzo della definizione alias del gestore code in questa illustrazione non ha selezionato un gestore code di destinazione differente. I messaggi che passano attraverso questo gestore code locale e indirizzati a 'QMB_priority' sono destinati al gestore code 'QMB'. Il nome del gestore code alias viene utilizzato per creare il flusso di messaggi separato.

Passaggio di messaggi attraverso il sistema

È possibile passare i messaggi attraverso il proprio sistema in tre modi: utilizzando il nome ubicazione, utilizzando un alias per il gestore code o selezionando una coda di trasmissione.

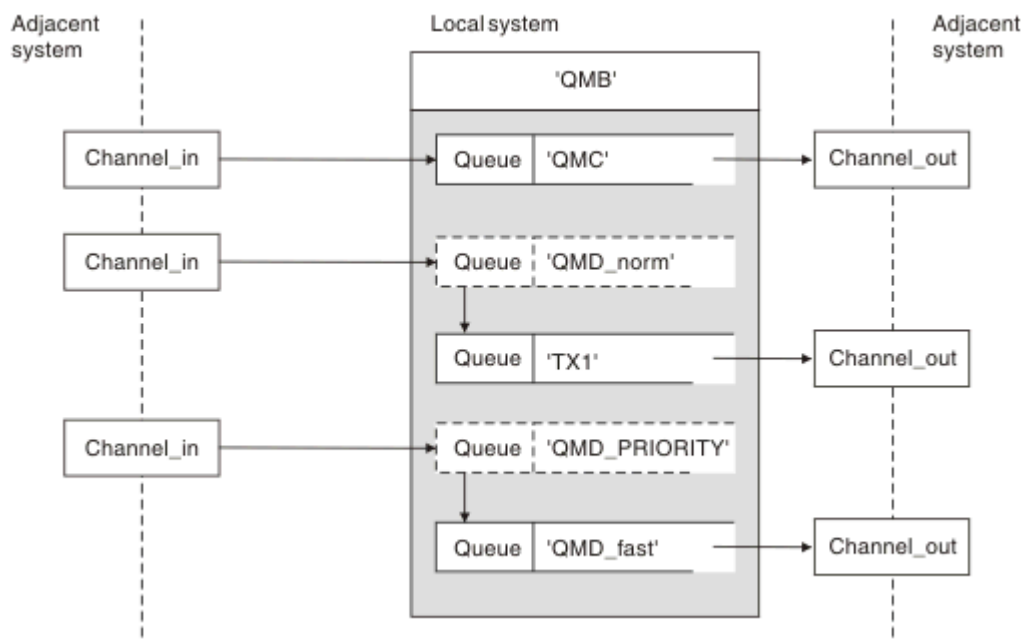


Figura 9. Tre metodi di trasmissione dei messaggi attraverso il sistema

La tecnica mostrata in [Figura 8 a pagina 210](#) in “ricezione di messaggi” a [pagina 210](#), ha mostrato come viene catturato un flusso alias. La [Figura 9 a pagina 211](#) illustra il modo in cui le reti vengono create, riunendo le tecniche precedentemente descritte.

La configurazione mostra un canale che consegna tre messaggi con destinazioni differenti:

1. QB alle QMC
2. QB alle QMD_norm
3. QB alle QMD_PRIORITY

È necessario passare il primo flusso di messaggi attraverso il sistema senza modifiche. È necessario passare il secondo flusso di messaggi attraverso una coda di trasmissione e un canale differenti. Per il secondo flusso di messaggi, è necessario anche risolvere i messaggi per il nome gestore code alias `QMD_norm` nel gestore code `QMD`. Il terzo flusso di messaggi sceglie una coda di trasmissione diversa senza altre modifiche.

In un ambiente cluster, i messaggi vengono trasmessi attraverso una coda di trasmissione cluster. Di solito, una singola coda di trasmissione, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, trasferisce tutti i messaggi a tutti i gestori code in tutti i cluster di cui è membro il gestore code; consultare [Un cluster di gestori code](#). È possibile definire code di trasmissione separate per tutti o per alcuni dei gestori code nei cluster di cui il gestore code è membro.

I seguenti metodi descrivono le tecniche applicabili a un ambiente di accodamento distribuito.

Utilizza questi metodi

Per queste configurazioni, è necessario preparare:

- Definizioni del canale di input
- Definizioni del canale di output
- Code di trasmissione:
 - `QMC`
 - `TX1`
 - `QMD_fast`
- Definizioni alias del gestore code:
 - `QMD_norm` con `QMD_norm` per `QMD` tramite `TX1`
 - `QMD_PRIORITY` con `QMD_PRIORITY` per `QMD_PRIORITY` tramite `QMD_fast`

Nota: Nessuno dei flussi di messaggi visualizzati nell'esempio modifica la coda di destinazione. Gli alias del nome gestore code forniscono la separazione dei flussi di messaggi.

Metodo 1: utilizzare il nome dell'ubicazione in entrata

Si riceveranno messaggi con un'intestazione di trasmissione contenente un altro nome di ubicazione, ad esempio `QMC`. La configurazione più semplice consiste nel creare una coda di trasmissione con tale nome, `QMC`. Il canale che servizi la coda di trasmissione distribuisce il messaggio non modificato alla destinazione successiva.

Metodo 2: utilizzare un alias per il gestore code

Il secondo metodo consiste nell'utilizzare la definizione dell'oggetto alias del Gestore code, ma specificare un nuovo nome di ubicazione, `QMD`, e una particolare coda di trasmissione, `TX1`. Questa azione:

- Termina il flusso di messaggi alias impostato dall'alias del nome del gestore code `QMD_norm`, ossia la classe di servizio denominata `QMD_norm`.
- Modifica le intestazioni di trasmissione su questi messaggi da `QMD_norm` a `QMD`.

Metodo 3: selezionare una coda di trasmissione

Il terzo metodo consiste nell'avere un oggetto alias del gestore code definito con lo stesso nome dell'ubicazione di destinazione, `QMD_PRIORITY`. Utilizzare la definizione di alias del gestore code per selezionare una coda di trasmissione particolare, `QMD_fast`, e quindi un altro canale. Le intestazioni di trasmissione su questi messaggi rimangono invariate.

Separazione dei flussi di messaggi

È possibile utilizzare un alias del gestore code per creare flussi di messaggi separati per inviare messaggi allo stesso gestore code.

Motivi per separare i messaggi in flussi di messaggi differenti

In un ambiente di accodamento distribuito, la necessità di separare i messaggi nello stesso gestore code in flussi di messaggi differenti può verificarsi per una serie di ragioni. Ad esempio:

- Potrebbe essere necessario fornire un flusso separato per i messaggi grandi, medi e piccoli. Questa necessità si applica anche in un ambiente di cluster e, in questo caso, è possibile creare cluster che si sovrappongono. Ci sono una serie di motivi per cui è possibile farlo, ad esempio:
 - Per consentire alle diverse organizzazioni di avere la propria amministrazione.
 - Per consentire la gestione separata delle applicazioni indipendenti.
 - Per creare una classe di servizio. Ad esempio, è possibile avere un cluster denominato STAFF che è un sottoinsieme del cluster denominato STUDENTI. Quando si inserisce un messaggio in una coda pubblicizzata nel cluster STAFF, viene utilizzato un canale limitato. Quando si inserisce un messaggio in una coda pubblicizzata nel cluster STUDENTI, è possibile utilizzare un canale generale o un canale limitato.
 - Per creare ambienti di test e di produzione.
- Potrebbe essere necessario instradare i messaggi in entrata in base a percorsi differenti dal percorso dei messaggi generati localmente.
- L'installazione potrebbe richiedere di pianificare lo spostamento dei messaggi in determinati orari (ad esempio, durante la notte) e i messaggi devono quindi essere memorizzati in code riservate fino a quando non vengono pianificati.

Flusso di messaggi di esempio

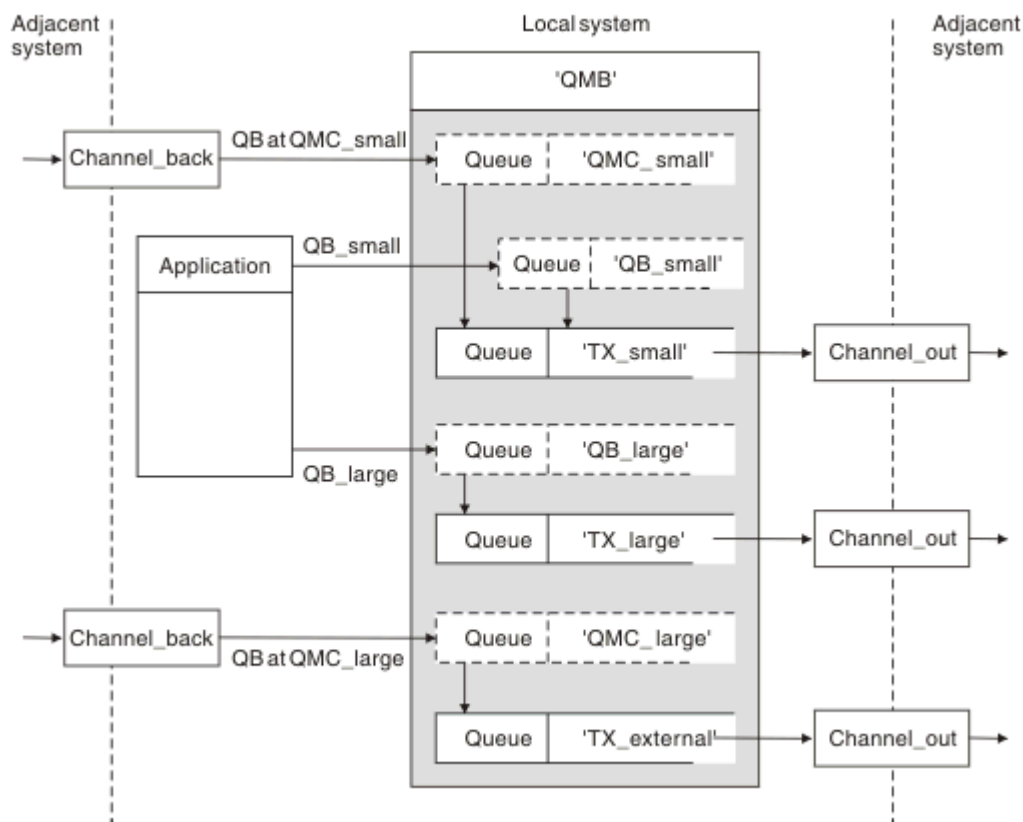


Figura 10. Separazione dei flussi di messaggi

Nell'esempio mostrato in [Figura 10](#) a pagina 214, i due flussi in entrata hanno come alias i nomi dei gestori code 'QMC_small' e 'QMC_large'. Fornire a questi flussi una definizione di alias del gestore code per catturare tali flussi per il gestore code locale. Si dispone di un'applicazione che si rivolge a due code remote ed è necessario che questi flussi di messaggi siano tenuti separati. Si forniscono due definizioni di coda remota che specificano la stessa posizione, 'QMC', ma specificano code di trasmissione differenti. Questa definizione mantiene separati i flussi e non è necessario alcun elemento aggiuntivo all'estremità poiché hanno lo stesso nome gestore code di destinazione nelle intestazioni di trasmissione. L'utente fornisce:

- Le definizioni di canale in ingresso
- Le due definizioni di coda remota QB_small e QB_large
- Le due definizioni di alias del gestore code QMC_small e QMC_large
- Le tre definizioni del canale di invio
- Tre code di trasmissione: TX_small, TX_large e TX_external

Coordinamento con sistemi adiacenti

Quando si utilizza un nome alternativo del gestore code per creare un flusso di messaggi separato, è necessario coordinare questa attività con l'amministratore di sistema all'estremità remota del canale di messaggi per assicurarsi che l'alias del gestore code corrispondente sia disponibile.

Concentrare i messaggi in diverse ubicazioni

È possibile concentrare i messaggi destinati a varie ubicazioni su un singolo canale.

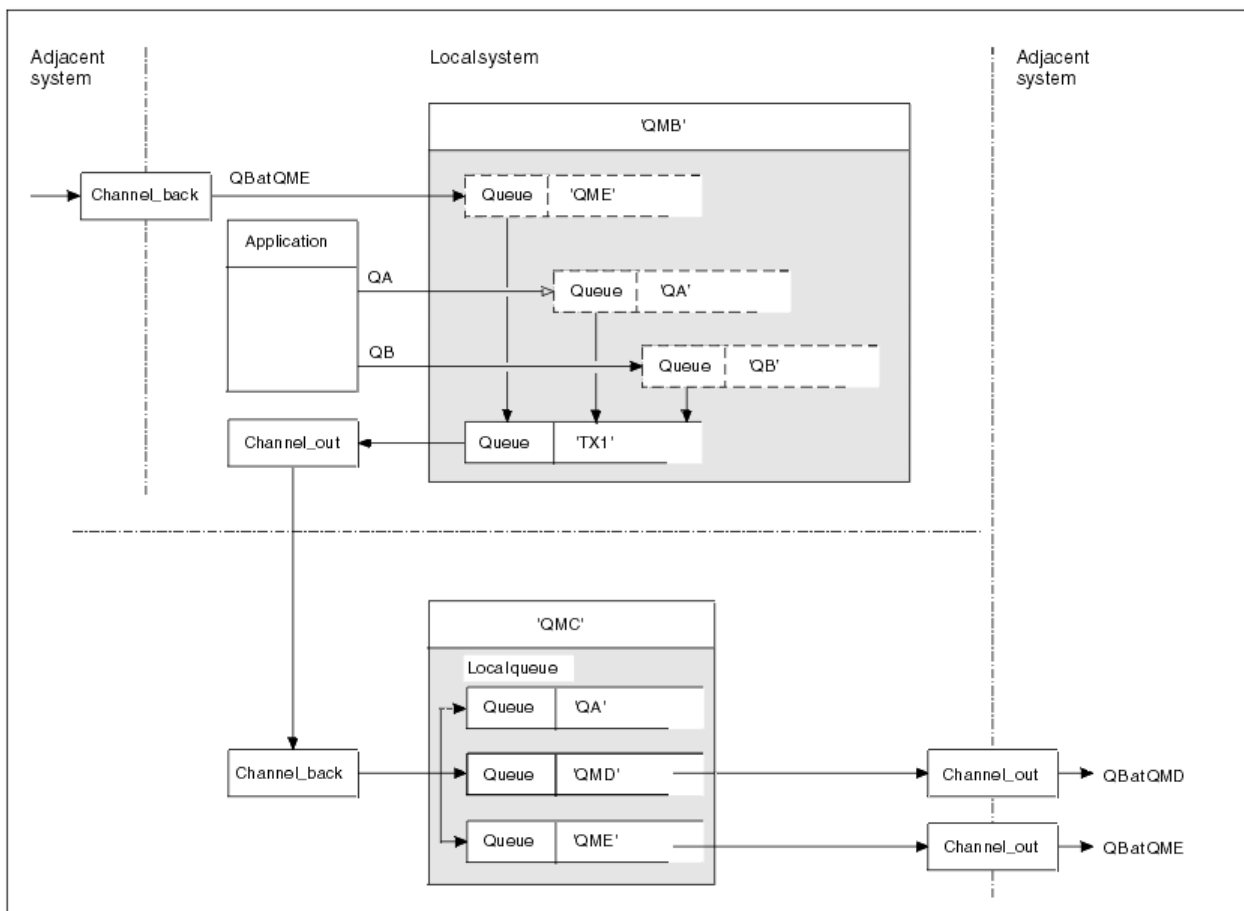


Figura 11. Combinazione dei flussi di messaggi su un canale

La Figura 11 a pagina 215 illustra una tecnica di accodamento distribuito per concentrare i messaggi destinati a varie ubicazioni su un canale. Due possibili utilizzi sono:

- Concentrazione del traffico di messaggi attraverso un gateway
- Utilizzo delle autostrade ad ampia larghezza di banda tra i nodi

In questo esempio, i messaggi provenienti da origini diverse, locali e adiacenti, con code di destinazione e gestori code differenti, vengono trasmessi attraverso la coda di trasmissione 'TX1' al gestore code QMC. Il gestore code QMC consegna i messaggi in base alle destinazioni. Uno è impostato su una coda di trasmissione 'QMD' per la trasmissione in avanti al gestore code QMD. Un altro è impostato su una coda di trasmissione 'QME' per la trasmissione successiva al gestore code QME. Altri messaggi vengono inseriti nella coda locale 'QA'.

È necessario fornire:

- Definizioni canale
- Coda di trasmissione TX1
- Definizioni coda remota:
 - QA con 'QA su QMC tramite TX1'
 - QB con 'QB a QMD tramite TX1'
- Definizione alias gestore code:
 - QME con 'QME tramite TX1'

L'amministratore complementare che sta configurando QMC deve fornire:

- Ricezione della definizione di canale con lo stesso nome canale

- QMD coda di trasmissione con definizione canale di invio associata
- QME coda di trasmissione con definizione canale di invio associata
- QA oggetto coda locale.

Deviazione dei flussi di messaggi in un'altra destinazione

È possibile ridefinire la destinazione di alcuni messaggi utilizzando gli alias del gestore code e le code di trasmissione.

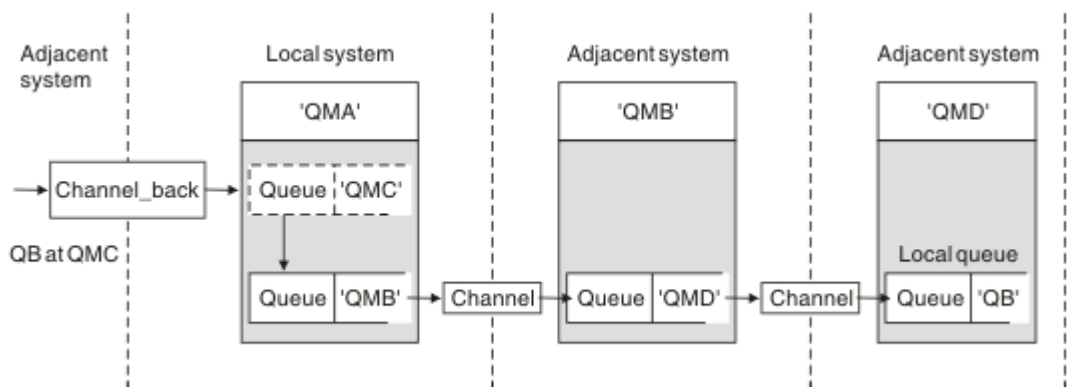


Figura 12. Deviazione dei flussi di messaggi in un'altra destinazione

Figura 12 a pagina 216 illustra come ridefinire la destinazione di determinati messaggi. I messaggi in entrata in QMA sono destinati a 'QB in QMC'. Normalmente arrivano a QMA e vengono posizionati su una coda di trasmissione chiamata QMC che è stata parte di un canale a QMC. QMA deve deviare i messaggi su QMD, ma è in grado di raggiungere QMD solo su QMB. Questo metodo è utile quando è necessario spostare un servizio da un'ubicazione all'altra e consentire ai sottoscrittori di continuare a inviare messaggi su base temporanea fino a quando non si sono adattati al nuovo indirizzo.

Il metodo di reinstradamento dei messaggi in entrata destinati per un determinato gestore code a un gestore code differente utilizza:

- Un alias del gestore code per modificare il gestore code di destinazione in un altro gestore code e per selezionare una coda di trasmissione per il sistema adiacente
- Una coda di trasmissione per servire il gestore code adiacente
- Una coda di trasmissione sul gestore code adiacente per l'instradamento verso il gestore code di destinazione

È necessario fornire:

- Definizione Channel_back
- QMC definizione oggetto alias del gestore code con QB da QMD a QMB
- Definizione Channel_out
- La coda di trasmissione associata QMB

L'amministratore complementare che sta configurando QMB deve fornire:

- La definizione channel_back corrispondente
- La coda di trasmissione, QMD
- La definizione canale associata a QMD

È possibile utilizzare gli alias all'interno di un ambiente cluster. Per informazioni, consultare [“Cluster e alias del gestore code”](#) a pagina 400.

Invio di messaggi a un elenco di distribuzione

Su Multiplatforms, è possibile utilizzare una singola chiamata MQPUT per fare in modo che un'applicazione invii un messaggio a diverse destinazioni.

In IBM MQ su Multiplatforms, un'applicazione può inviare un messaggio a più destinazioni con una singola chiamata MQPUT. È possibile eseguire questa operazione sia in un ambiente di accodamento distribuito che in un ambiente cluster. È necessario definire le destinazioni in un elenco di distribuzione, come descritto in [Elenchi di distribuzione](#).

Non tutti i gestori code supportano gli elenchi di distribuzione. Quando un MCA stabilisce una connessione con un partner, determina se il partner supporta gli elenchi di distribuzione e imposta di conseguenza un indicatore sulla coda di trasmissione. Se un'applicazione tenta di inviare un messaggio destinato ad un elenco di distribuzione ma il partner non supporta gli elenchi di distribuzione, l'MCA mittente intercetta il messaggio e lo inserisce nella coda di trasmissione una volta per ciascuna destinazione prevista.

Un MCA di ricezione garantisce che i messaggi inviati a un elenco di distribuzione vengano ricevuti in modo sicuro in tutte le destinazioni previste. Se qualche destinazione ha esito negativo, l'MCA stabilisce quali destinazioni hanno avuto esito negativo. Può quindi generare report di eccezioni e provare a inviare nuovamente i messaggi.

Coda di risposta

È possibile creare un loop di elaborazione coda remota completo utilizzando una coda di risposta.

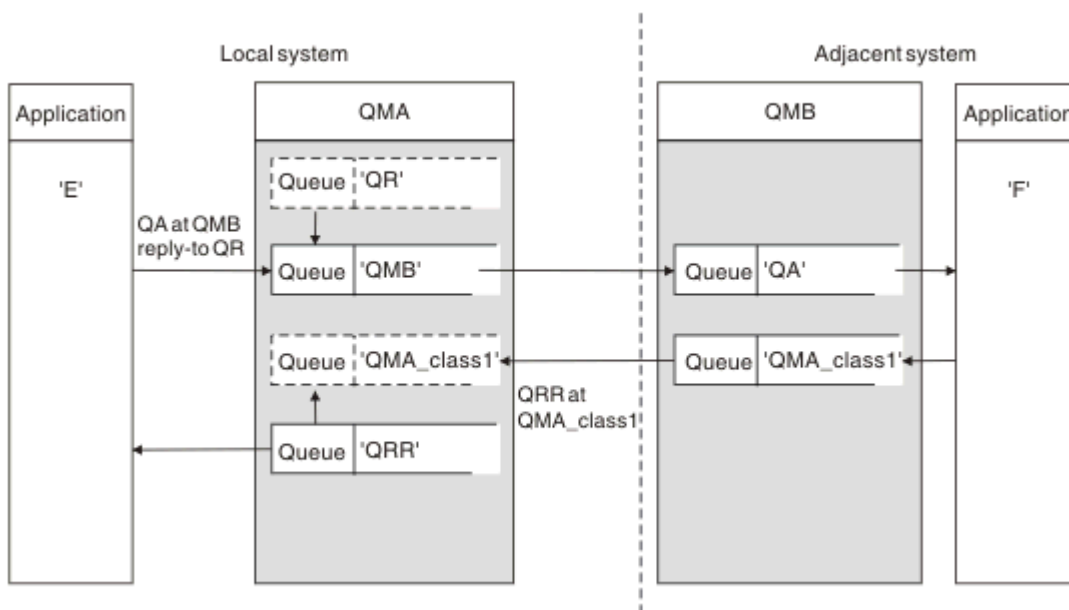


Figura 13. Sostituzione del nome della coda di risposta durante la chiamata PUT

Un loop di elaborazione della coda remota completo che utilizza una coda di risposta viene visualizzato in [Figura 13 a pagina 217](#). Questo loop si applica sia in un ambiente di accodamento distribuito che in un ambiente cluster. I dettagli sono come mostrato in [Tabella 21 a pagina 224](#).

L'applicazione apre QA a QMB e inserisce i messaggi su tale coda. Ai messaggi viene fornito un nome coda di risposta QR, senza specificare il nome del gestore code. QMA del gestore code trova l'oggetto coda di risposta QR ed estrae da esso il nome alias di QRR e il nome gestore code QMA_class1. Questi nomi vengono inseriti nei campi di risposta dei messaggi.

I messaggi di risposta dalle applicazioni in QMB vengono indirizzati a QRR in QMA_class1. La definizione del nome alias del gestore code QMA_class1 viene utilizzata dal gestore code per trasmettere i messaggi a se stesso e alla coda QRR.

Questo scenario illustra il modo in cui le applicazioni possono scegliere una classe di servizio per i messaggi di risposta. La classe è implementata dalla coda di trasmissione QMA_class1 in QMB, insieme

alla definizione dell'alias del gestore code, QMA_class1 in QMA. In questo modo, è possibile modificare la coda di risposta dell'applicazione in modo che i flussi siano separati senza coinvolgere l'applicazione. L'applicazione sceglie sempre QR per questa particolare classe di servizi. È possibile modificare la classe di servizi con la definizione della coda di risposta QR.

È necessario creare:

- QR definizione coda di risposta
- QMB oggetto coda di trasmissione
- Definizione Channel_out
- Definizione Channel_back
- Definizione alias gestore code QMA_class1
- Oggetto coda locale QRR, se non esiste

L'amministratore complementare sul sistema adiacente deve creare:

- Definizione di canale di ricezione
- Oggetto coda di trasmissione QMA_class1
- Canale di invio associato
- QA oggetto coda locale.

I programmi applicativi utilizzano:

- Nome coda di risposta a QR nelle chiamate di inserimento
- Nome coda QRR nelle chiamate get

In questo modo, è possibile modificare la classe di servizio come necessario, senza coinvolgere l'applicazione. È possibile modificare l'alias di risposta 'QR', insieme alla coda di trasmissione 'QMA_class1' e all'alias del gestore code 'QMA_class1'.

Se non viene trovato alcun oggetto alias di risposta quando il messaggio viene inserito nella coda, il nome del gestore code locale viene inserito nel campo del nome del gestore code di risposta vuoto. Il nome della coda di risposta rimane invariato.

Limitazione risoluzione nome

Poiché la risoluzione del nome è stata eseguita per la coda di risposta in 'QMA' quando è stato inserito il messaggio originale, non è consentita alcuna ulteriore risoluzione del nome in 'QMB'. Il messaggio viene inserito con il nome fisico della coda reply - to dall'applicazione che risponde.

Le applicazioni devono essere consapevoli che il nome che utilizzano per la coda di risposta è diverso dal nome della coda effettiva in cui si trovano i messaggi di ritorno.

Ad esempio, quando due classi di servizio vengono fornite per l'utilizzo di applicazioni con nomi alias coda reply - to 'C1_alias' e 'C2_alias', le applicazioni utilizzano questi nomi come nomi coda reply - to nelle chiamate di inserimento messaggi. Tuttavia, le applicazioni in realtà prevedono che i messaggi vengano visualizzati nelle code 'C1' per 'C1_alias' e 'C2' per 'C2_alias'.

Tuttavia, un'applicazione è in grado di effettuare una chiamata di interrogazione sulla coda alias di risposta per controllare il nome della coda reale che deve utilizzare per ottenere i messaggi di risposta.

Concetti correlati

[“Come creare il gestore code e gli alias di risposta” a pagina 207](#)

Questo argomento illustra i tre modi in cui è possibile creare una definizione di coda remota.

[“Esempio di alias della coda reply - to” a pagina 219](#)

Questo esempio illustra l'utilizzo di un alias reply - to per selezionare un instradamento differente (coda di trasmissione) per i messaggi restituiti. L'utilizzo di questa funzionalità richiede la modifica del nome della coda di risposta in collaborazione con le applicazioni.

[“Funzionamento dell'esempio” a pagina 220](#)

Una spiegazione dell'esempio e del modo in cui il gestore code utilizza l'alias della coda di risposta.

“Procedura dettagliata per l'alias della coda di risposta” a pagina 221

Una procedura dettagliata del processo da un'applicazione che immette un messaggio su una coda remota alla stessa applicazione che rimuove il messaggio di risposta dalla coda di risposta alias.

Esempio di alias della coda reply - to

Questo esempio illustra l'utilizzo di un alias reply - to per selezionare un instradamento differente (coda di trasmissione) per i messaggi restituiti. L'utilizzo di questa funzionalità richiede la modifica del nome della coda di risposta in collaborazione con le applicazioni.

Come mostrato in Figura 14 a pagina 219, l'instradamento di ritorno deve essere disponibile per i messaggi di risposta, inclusi la coda di trasmissione, il canale e l'alias del gestore code.

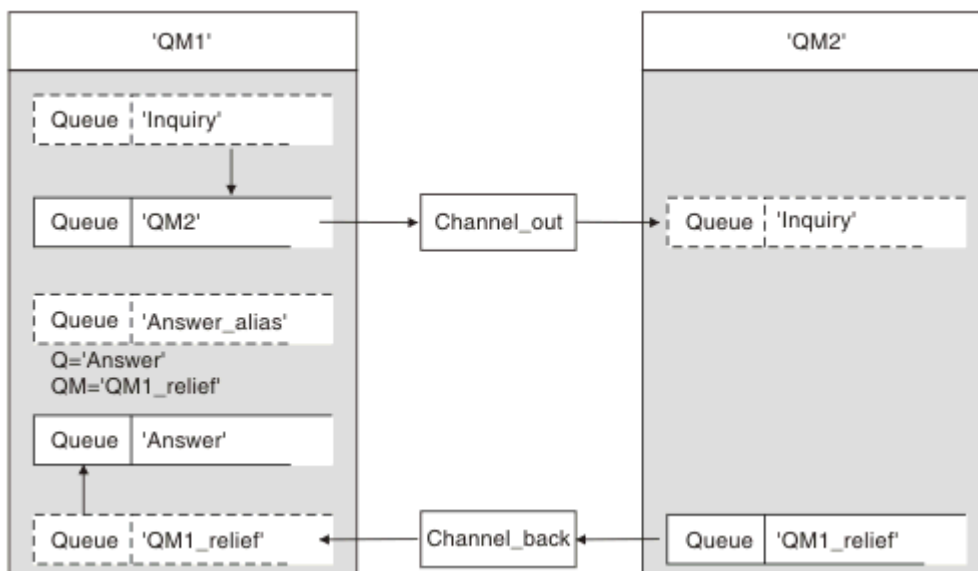


Figura 14. Esempio di alias della coda reply - to

Questo esempio è per le applicazioni del richiedente in 'QM1' che inviano messaggi alle applicazioni server in 'QM2'. I messaggi sul server devono essere restituiti tramite un canale alternativo utilizzando la coda di trasmissione 'QM1_relief' (il canale di ritorno predefinito viene servito con una coda di trasmissione 'QM1').

L'alias della coda di risposta è un utilizzo particolare della definizione della coda remota denominata 'Answer_alias'. Le applicazioni in QM1 includono questo nome, 'Answer_alias', nel campo di risposta di tutti i messaggi che inseriscono nella coda 'Inquiry'.

La definizione della coda di risposta 'Answer_alias' è definita come 'Risposta a QM1_relief'. Le applicazioni in QM1 prevedono che le loro risposte vengano visualizzate nella coda locale denominata 'Risposta'.

Le applicazioni server in QM2 utilizzano il campo di risposta dei messaggi ricevuti per ottenere i nomi della coda e del gestore code per i messaggi di risposta al richiedente in QM1.

Definizioni utilizzate in questo esempio in QM1

L'amministratore di sistema IBM MQ in QM1 deve assicurarsi che la coda di risposta 'Risposta' sia creata insieme agli altri oggetti. Il nome dell'alias del gestore code, contrassegnato con un '*', deve essere in accordo con il nome del gestore code nella definizione dell'alias della coda di risposta, contrassegnato anche con un '*'.

Oggetto	Definizione	
Coda di trasmissione locale	QM2	
Definizione di coda remota	Nome oggetto	Interrogazione
	Nome gestore code remoto	QM2

Oggetto	Definizione	
	Nome coda remota	Interrogazione
	Nome coda di trasmissione	QM2 (PREDEFINITO)
Alias gestore code	Nome oggetto	QM1_relief *
	Nome del gestore code	QM1
	Nome coda	(vuoto)
Alias coda di risposta	Nome oggetto	Alias_risposta
	Nome gestore code remoto	QM1_relief *
	Nome coda remota	Risposta

Inserisci definizione in QM1

Le applicazioni compilano i campi di risposta con il nome alias della coda di risposta e lasciano vuoto il campo del nome del gestore code.

Campo	Contenuto
Nome coda	Interrogazione
Nome del gestore code	(vuoto)
Nome delle repliche alla coda	Alias_risposta
Gestore code di risposta	(vuoto)

Definizioni utilizzate in questo esempio in QM2

L'amministratore di sistema IBM MQ in QM2 deve assicurarsi che la coda locale esista per i messaggi in entrata e che la coda di trasmissione denominata correttamente sia disponibile per i messaggi di risposta.

Oggetto	Definizione
Coda locale	Interrogazione
Coda di trasmissione	QM1_relief

Inserire la definizione in QM2

Le applicazioni in QM2 richiamano il nome della coda di risposta e il nome del gestore code dal messaggio originale e li utilizzano durante l'inserimento del messaggio di risposta nella coda di risposta.

Campo	Contenuto
Nome coda	Risposta
Nome del gestore code	QM1_relief

Funzionamento dell'esempio

Una spiegazione dell'esempio e del modo in cui il gestore code utilizza l'alias della coda di risposta.

In questo esempio, le applicazioni del richiedente in QM1 utilizzano sempre 'Answer_alias' come coda di risposta nel relativo campo della chiamata put. Richiamano sempre i messaggi dalla coda denominata 'Risposta'.

Le definizioni degli alias della coda di risposta sono disponibili per l'utilizzo da parte dell'amministratore di sistema QM1 per modificare il nome della coda di risposta 'Risposta' e dell'instradamento di ritorno 'QM1_relief'.

La modifica del nome della coda 'Risposta' di solito non è utile perché le applicazioni QM1 si aspettano le risposte in questa coda. Tuttavia, l'amministratore di sistema QM1 è in grado di modificare il percorso di ritorno (classe di servizio), se necessario.

Modalità con cui il gestore code utilizza l'alias della coda di risposta

Il gestore code QM1 richiama le definizioni dall'alias della coda di risposta quando il nome della coda di risposta, incluso nella chiamata di inserimento dall'applicazione, è uguale all'alias della coda di risposta e la parte del gestore code è vuota.

Il gestore code sostituisce il nome della coda di risposta nella chiamata di inserimento con il nome della coda dalla definizione. Sostituisce il nome del gestore code vuoto nella chiamata di inserimento con il nome del gestore code dalla definizione.

Questi nomi vengono portati con il messaggio nel descrittore del messaggio.

<i>Tabella 18. Alias coda di risposta</i>		
Nome campo	Inserisci chiamata	Intestazione trasmissione
Nome delle repliche alla coda	Alias_risposta	Risposta
Nome gestore code di risposta	(vuoto)	QM1_relief

Procedura dettagliata per l'alias della coda di risposta

Una procedura dettagliata del processo da un'applicazione che immette un messaggio su una coda remota alla stessa applicazione che rimuove il messaggio di risposta dalla coda di risposta alias.

Per completare questo esempio, guardiamo il processo.

1. L'applicazione apre una coda denominata 'Inquiry' e vi inserisce i messaggi. L'applicazione imposta i campi reply - to del descrittore del messaggio su:

Nome delle repliche alla coda	Alias_risposta
Nome gestore code di risposta	(vuoto)

2. Il gestore code 'QM1' risponde al nome del gestore code vuoto controllando la definizione di una coda remota con il nome 'Answer_alias'. In caso contrario, il gestore code inserisce il proprio nome, 'QM1', nel campo del gestore code di risposta del descrittore del messaggio.
3. Se il gestore code trova una definizione di coda remota con il nome 'Answer_alias', estrae il nome coda e i nomi gestore code dalla definizione (nome coda = 'Risposta' e nome gestore code = 'QM1_relief'). Quindi, li inserisce nei campi reply - to del descrittore del messaggio.
4. Il gestore code 'QM1' utilizza la definizione della coda remota 'Inquiry' per stabilire che la coda di destinazione prevista si trova sul gestore code 'QM2' e che il messaggio viene inserito nella coda di trasmissione 'QM2'. 'QM2' è il nome della coda di trasmissione predefinita per i messaggi destinati alle code sul gestore code 'QM2'.
5. Quando il gestore code 'QM1' inserisce il messaggio nella coda di trasmissione, aggiunge un'intestazione di trasmissione al messaggio. Questa intestazione contiene il nome della coda di destinazione, 'Inquiry', e il gestore code di destinazione, 'QM2'.
6. Il messaggio arriva al gestore code 'QM2' e viene inserito nella coda locale 'Inquiry'.
7. Un'applicazione richiama il messaggio da questa coda ed elabora il messaggio. L'applicazione prepara un messaggio di risposta e inserisce questo messaggio di risposta nel nome della coda di risposta dal descrittore del messaggio originale:

Nome delle repliche alla coda	Risposta
Nome gestore code di risposta	QM1_relief

8. Il gestore code 'QM2' esegue il comando put. Rilevando che il nome del gestore code 'QM1_relief' è un gestore code remoto, posiziona il messaggio sulla coda di trasmissione con lo stesso nome,

'QM1_relief'. Al messaggio viene fornita un'intestazione di trasmissione contenente il nome della coda di destinazione, 'Risposta', e il gestore code di destinazione, 'QM1_relief'.


9. Il messaggio viene trasferito al gestore code QM1'. Il gestore code, riconosce che il nome del gestore code 'QM1_relief' è un alias, estrae dalla definizione alias 'QM1_relief' il nome del gestore code fisico 'QM1'.
10. Il gestore code 'QM1' inserisce il messaggio nel nome della coda contenuto nell'intestazione di trasmissione, 'Risposta'.
11. L'applicazione estrae il messaggio di risposta dalla coda 'Risposta'.

Considerazioni sulla rete

In un ambiente di accodamento distribuito, poiché le destinazioni dei messaggi vengono indirizzate solo con un nome coda e un nome gestore code, si applicano alcune regole.

1. Dove viene fornito il nome del gestore code e il nome è diverso dal nome del gestore code locale:
 - Una coda di trasmissione deve avere lo stesso nome. Questa coda di trasmissione deve far parte di un canale di messaggi che sposta i messaggi in un altro gestore code oppure
 - È necessario che esista una definizione dell'alias del gestore code per risolvere il nome del gestore code nello stesso nome o in un altro nome del gestore code e nella coda di trasmissione facoltativa oppure
 - Se il nome della coda di trasmissione non può essere risolto ed è stata definita una coda di trasmissione predefinita, viene utilizzata la coda di trasmissione predefinita.
2. Dove viene fornito solo il nome della coda, una coda di qualsiasi tipo ma con lo stesso nome deve essere disponibile sul gestore code locale. Questa coda può essere una definizione di coda remota che si risolve in: una coda di trasmissione a un gestore code adiacente, un nome gestore code e una coda di trasmissione facoltativa.

Per vedere come funziona in un ambiente di cluster, consultare [Cluster](#).

 Se i gestori code sono in esecuzione in un QSG (queue sharing group) e IGQ (intra - group queuing) è abilitato, è possibile utilizzare SYSTEM.QSG.TRANSMIT.QUEUE. Per ulteriori informazioni, consultare [Intra - group queuing](#).

Si consideri lo scenario di un canale di messaggi che sposta i messaggi da un gestore code ad un altro in un ambiente di accodamento distribuito.

I messaggi spostati sono stati originati da qualsiasi altro gestore code nella rete e potrebbero arrivare alcuni messaggi con un nome gestore code sconosciuto come destinazione. Questo problema può verificarsi quando un nome gestore code è stato modificato o è stato rimosso dal sistema, ad esempio.

Il programma del canale riconosce questa situazione quando non riesce a trovare una coda di trasmissione per questi messaggi e inserisce i messaggi nella coda dei messaggi non recapitati (messaggi non recapitati). È tua responsabilità cercare questi messaggi e fare in modo che vengano inoltrati alla destinazione corretta. In alternativa, restituirli all'originatore, dove l'originatore può essere accertato.

I report di eccezione vengono generati in queste circostanze, se i messaggi di report sono stati richiesti nel messaggio originale.

Convenzione di risoluzione dei nomi

La risoluzione dei nomi che modifica l'identità della coda di destinazione (ossia, la modifica del nome da logico a fisico), si verifica solo una volta e solo sul gestore code di origine.

L'utilizzo successivo delle varie possibilità alias deve essere utilizzato solo quando si separano e si combinano i flussi di messaggi.

Instradamento di ritorno

I messaggi possono contenere un indirizzo di ritorno nel formato del nome di una coda e di un gestore code. Questo modulo di indirizzo di ritorno può essere utilizzato sia in un ambiente di accodamento distribuito che in un ambiente cluster.

Questo indirizzo viene normalmente specificato dall'applicazione che crea il messaggio. Può essere modificato da qualsiasi applicazione che gestisce il messaggio, incluse le applicazioni di uscita utente.

Indipendentemente dall'origine di questo indirizzo, qualsiasi applicazione che gestisce il messaggio potrebbe scegliere di utilizzare questo indirizzo per restituire i messaggi di risposta, di stato o di report all'applicazione di origine.

Il modo in cui vengono instradati questi messaggi di risposta non è diverso dal modo in cui viene instradato il messaggio originale. È necessario tenere presente che i flussi di messaggi creati per altri gestori code richiedono flussi di ritorno corrispondenti.

Conflitti di nomi fisici

Il nome della coda di risposta di destinazione è stato risolto in un nome di coda fisica sul gestore code originale. Non deve essere risolto nuovamente nel gestore code di risposta.

È una probabile possibilità per i problemi di conflitto dei nomi che possono essere evitati solo da un accordo a livello di rete sui nomi delle code fisiche e logiche.

Gestione delle traduzioni dei nomi delle code

Quando si crea una definizione dell'alias del gestore code o una definizione della coda remota, la risoluzione del nome viene eseguita per ogni messaggio che porta quel nome. Questa situazione deve essere gestita.

Questa descrizione viene fornita per progettisti di applicazioni e pianificatori di canali interessati a un singolo sistema che dispone di canali di messaggi per sistemi adiacenti. Prende una visione locale di pianificazione e controllo del canale.

Quando si crea una definizione dell'alias del gestore code o una definizione della coda remota, la risoluzione del nome viene eseguita per ogni messaggio che contiene tale nome, indipendentemente dall'origine del messaggio. Per sovrintendere a questa situazione, che potrebbe coinvolgere un numero elevato di code in una rete di gestori code, è necessario tenere le seguenti tabelle:

- I nomi delle code di origine e dei gestori code di origine rispetto ai nomi delle code risolti, ai nomi dei gestori code risolti e ai nomi delle code di trasmissione risolti, con il metodo di risoluzione
- I nomi delle code di origine rispetto a:
 - Nomi delle code di destinazione risolte
 - Nomi dei gestori code di destinazione risolti
 - Code di trasmissione
 - Nomi dei canali di messaggi
 - Nomi di sistema adiacenti
 - Nomi coda di risposta

Nota: L'utilizzo del termine *origine* in questo contesto fa riferimento al nome della coda o al nome del gestore code fornito dall'applicazione o a un programma del canale durante l'apertura di una coda per l'inserimento di messaggi.

Un esempio di ciascuna di queste tabelle viene mostrato in [Tabella 19 a pagina 224](#), [Tabella 20 a pagina 224](#) e [Tabella 21 a pagina 224](#).

I nomi in queste tabelle sono derivati dagli esempi in questa sezione e questa tabella non è intesa come un esempio pratico di risoluzione dei nomi di coda in un nodo.

Tabella 19. Risoluzione nome coda nel gestore code QMA

Coda di origine specificata quando la coda è aperta	Gestore code di origine specificato quando la coda è aperta	Nome coda risolto	Nome del gestore code risolto	Nome coda di trasmissione risolta	Tipo di risoluzione
QA_norm	-	QA_norm	QMB	QMB	Coda remota
(qualsiasi)	QMB	-	-	QMB	(nessuno)
QA_norm	-	QA_norm	QMB	TX1	Coda remota
QB	MMC	QB	MMD	QMB	Alias gestore code

Tabella 20. Risoluzione del nome coda nel gestore code QMB

Coda di origine specificata quando la coda è aperta	Gestore code di origine specificato quando la coda è aperta	Nome coda risolto	Nome del gestore code risolto	Nome coda di trasmissione risolta	Tipo di risoluzione
QA_norm	-	QA_norm	QMB	-	(nessuno)
QA_norm	QMB	QA_norm	QMB	-	(nessuno)
QA_norm	PRIORITÀ_QM	QA_norm	QMB	-	Alias gestore code
(qualsiasi)	MMC	(qualsiasi)	MMC	MMC	(nessuno)
(qualsiasi)	QMD_norma	(qualsiasi)	QMD_norma	TX1	Alias gestore code
(qualsiasi)	QMD_PRIORITY	(qualsiasi)	QMD_PRIORITY	QMD_veloce	Alias gestore code
(qualsiasi)	QMC_piccolo	(qualsiasi)	QMC_piccolo	TX_piccolo	Alias gestore code
(qualsiasi)	QMC_grande	(qualsiasi)	QMC_grande	TX_esterno	Alias gestore code
QB_piccolo	MMC	QB_piccolo	MMC	TX_piccolo	Coda remota
QB_grande	MMC	QB_grande	MMC	TX_grande	Coda remota
(qualsiasi)	DME	(qualsiasi)	DME	TX1	Alias gestore code
QA	MMC	QA	MMC	TX1	Coda remota
QB	MMD	QB	MMD	TX1	Coda remota

Tabella 21. Conversione del nome della coda di risposta nel gestore code QMA

Progettazione dell'applicazione		Definizione alias di risposta	
QMGR locale	Nome coda per i messaggi	Nome alias coda di risposta	Ridefinito in
QMA	RQ	Q.	QRR a QMA_class1

Numerazione sequenza messaggi canale

Il canale utilizza i numeri di sequenza per controllare che i messaggi siano consegnati nello stesso ordine in cui vengono presi dalla coda di trasmissione.

I numeri di sequenza del canale vengono controllati quando un canale viene avviato e se si verifica una mancata corrispondenza, ciò implica che i dati di sincronia persistenti sono stati persi su entrambi i lati del canale; ad esempio, una configurazione di ripristino di emergenza (DR) o la fine dell'elaborazione batch è stata interrotta quando il canale era in dubbio.

Reimpostando o ignorando le mancate corrispondenze del numero di sequenza, consultare **IgnoreSeqNumberMismatch** nella stanza *Channels del file qm.ini*, non rischia di perdere o duplicare un batch di messaggi e non reimposta lo stato in dubbio di un canale.

Queste informazioni possono essere visualizzate utilizzando **DISPLAY CHSTATUS**. Il numero di sequenza e un identificativo denominato LUWID vengono memorizzati nella memoria persistente per l'ultimo messaggio trasferito in batch. Questi valori vengono utilizzati durante l'avvio del canale per garantire che entrambe le estremità del collegamento concordino su quali messaggi sono stati trasferiti correttamente.

Richiamo sequenziale dei messaggi

Se un'applicazione inserisce una sequenza di messaggi nella stessa coda di destinazione, tali messaggi possono essere richiamati in sequenza da un'applicazione **singola** con una sequenza di operazioni MQGET, se sono soddisfatte le seguenti condizioni:

- Tutte le richieste di inserimento sono state effettuate dalla stessa applicazione.
- Tutte le richieste di inserimento provenivano dalla stessa unità di lavoro oppure tutte le richieste di inserimento sono state effettuate al di fuori di un'unità di lavoro.
- Tutti i messaggi hanno la stessa priorità.
- I messaggi hanno tutti la stessa persistenza.
- Per l'accodamento remoto, la configurazione è tale che può esistere solo un percorso dall'applicazione che effettua la richiesta di inserimento, attraverso il gestore code, attraverso l'intercomunicazione, al gestore code di destinazione e alla coda di destinazione.
- I messaggi non vengono inseriti in una coda di messaggi non instradabili (ad esempio, se una coda è temporaneamente piena).
- L'applicazione che riceve il messaggio non modifica deliberatamente l'ordine di richiamo, ad esempio specificando un particolare *MsgId* o *CorrelId* o utilizzando le priorità del messaggio.
- Solo un'applicazione sta eseguendo operazioni get per richiamare i messaggi dalla coda di destinazione. Se è presente più di un'applicazione, queste applicazioni devono essere progettate per ottenere tutti i messaggi in ogni sequenza inseriti da un'applicazione di invio.

Nota: I messaggi provenienti da altre attività e unità di lavoro potrebbero essere intervallati dalla sequenza, anche quando la sequenza è stata inserita all'interno di una singola unità di lavoro.

Se queste condizioni non possono essere soddisfatte e l'ordine dei messaggi sulla coda di destinazione è importante, l'applicazione può essere codificata per utilizzare il proprio numero di sequenza dei messaggi come parte del messaggio per assicurare l'ordine dei messaggi.

Sequenza di richiamo dei messaggi veloci e non persistenti

I messaggi non persistenti su un canale veloce potrebbero superare i messaggi persistenti sullo stesso canale e quindi arrivare fuori sequenza. L'MCA ricevente inserisce immediatamente i messaggi non persistenti nella coda di destinazione e li rende visibili. I messaggi persistenti non saranno visibili fino al punto di sincronizzazione successivo.

Test di loopback

Il *test di loopback* è una tecnica su Multiplatforms che consente di testare un collegamento di comunicazione senza collegarsi effettivamente ad un'altra macchina.

Si imposta una connessione tra due gestori code come se fossero su macchine separate, ma si verifica la connessione eseguendo un loop su un altro processo sulla stessa macchina. Questa tecnica significa che è possibile verificare il codice di comunicazione senza richiedere una rete attiva.

Il modo in cui si fa ciò dipende da quali prodotti e protocolli si stanno utilizzando.

Su sistemi Windows , è possibile utilizzare l'adattatore "loopback".

Per ulteriori informazioni, fare riferimento alla documentazione dei prodotti in uso.

Traccia del percorso e registrazione dell'attività

È possibile confermare l'instradamento di un messaggio attraverso una serie di gestori code in due modi.

È possibile utilizzare IBM MQ l'applicazione di visualizzazione dell'instradamento, disponibile tramite il comando di controllo **dspmqrte**, oppure è possibile utilizzare la registrazione dell'attività. Entrambi questi argomenti sono descritti in [Riferimento monitoraggio](#).

Introduzione alla gestione delle code distribuite

DQM (Distributed Queue Management) viene utilizzato per definire e controllare le comunicazioni tra gestori code.

Gestione code distribuite:




- Consente di definire e controllare i canali di comunicazione tra i gestori code
- Fornisce un servizio del canale dei messaggi per spostare i messaggi da un tipo di *coda locale*, nota come coda di trasmissione, a link di comunicazioni su un sistema locale e da link di comunicazioni a code locali su un gestore code di destinazione
- Fornisce funzioni per monitorare il funzionamento dei canali e diagnosticare i problemi, utilizzando pannelli, comandi e programmi

Le definizioni dei canali associano i nomi dei canali alle code di trasmissione, agli identificatori dei collegamenti di comunicazione e agli attributi dei canali. Le definizioni di canale sono implementate in modi diversi su piattaforme diverse. L'invio e la ricezione di messaggi è controllata da programmi noti come *agent canale messaggi* (MCA), che utilizzano le definizioni del canale per avviare e controllare la comunicazione.





Gli MCA a loro volta sono controllati dallo stesso DQM. La struttura è dipendente dalla piattaforma, ma in genere include listener e controlli trigger, insieme a comandi e pannelli dell'operatore.

Un *canale di messaggi* è un pipe unidirezionale per spostare i messaggi da un gestore code a un altro. Quindi un canale di messaggi ha due endpoint, rappresentati da una coppia di MCA. Ogni endpoint ha una definizione della sua fine del canale di messaggi. Ad esempio, un'estremità definirebbe un mittente, l'altra un destinatario.

Per i dettagli su come definire i canali, consultare:

-  [“Monitoraggio e controllo dei canali su AIX, Linux, and Windows” a pagina 258](#)
-  [“Monitoring and controlling channels on z/OS” a pagina 965](#)
-  [“Monitoraggio e controllo dei canali su IBM i” a pagina 282](#)

Per esempi di pianificazione del canale di messaggi, consultare:

-  [Esempio di pianificazione del canale dei messaggi per AIX, Linux, and Windows](#)
-  [Esempio di pianificazione del canale dei messaggi per IBM i](#)
-  [Esempio di pianificazione del canale dei messaggi per z/OS](#)
-  [Esempio di pianificazione del canale dei messaggi per z/OS utilizzo di gruppi di condivisione code](#)

Per informazioni sulle uscite canale, vedere [Programmi di uscita canale per canali di messaggistica](#).

Concetti correlati

[“Invio e ricezione di messaggi” a pagina 227](#)

La seguente figura mostra il modello di gestione della coda distribuita, dettagliando le relazioni tra entità quando vengono trasmessi i messaggi. Mostra anche il flusso per il controllo.

[“Funzione di controllo canale” a pagina 235](#)

La funzione di controllo del canale consente di definire, monitorare e controllare i canali.

[“Cosa succede quando un messaggio non può essere consegnato?” a pagina 249](#)

Quando un messaggio non può essere consegnato, l'MCA può elaborarlo in diversi modi. Può riprovare, può tornare al mittente o può inserirlo nella coda di messaggi non recapitabili.

[“File di inizializzazione e di configurazione” a pagina 254](#)

La gestione dei dati di inizializzazione del canale dipende dalla piattaforma IBM MQ .

[“Conversione dati per i messaggi” a pagina 255](#)

I messaggi IBM MQ potrebbero richiedere la conversione dei dati quando vengono inviati tra le code su gestori code differenti.

[“Scrittura dei propri agent del canale dei messaggi” a pagina 255](#)

IBM MQ consente di scrivere i propri programmi MCA (message channel agent) o di installarne uno da un fornitore di software indipendente.

[“Altre cose da considerare per la gestione della coda distribuita” a pagina 256](#)

Altri argomenti da considerare quando si prepara IBM MQ per la gestione delle code distribuite. Questo argomento riguarda le code di messaggi non recapitati, le code in uso, le estensioni di sistema e i programmi di uscita utente e i canali e i listener in esecuzione come applicazioni attendibili.

Riferimenti correlati

[Informazioni di configurazione di esempio](#)

Invio e ricezione di messaggi

La seguente figura mostra il modello di gestione della coda distribuita, dettagliando le relazioni tra entità quando vengono trasmessi i messaggi. Mostra anche il flusso per il controllo.

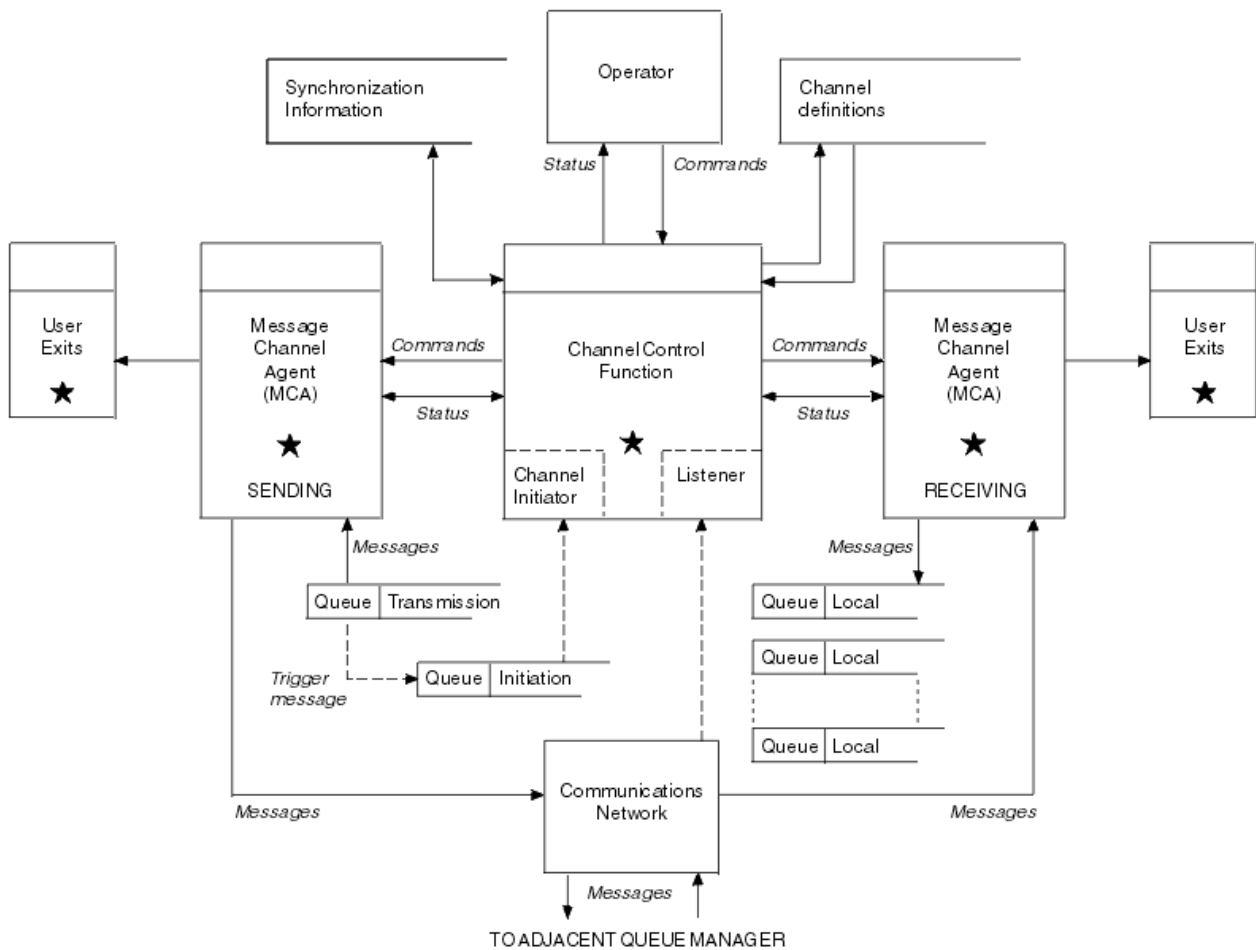


Figura 15. Modello di gestione code distribuite

Nota:

1. Esiste un MCA per canale, a seconda della piattaforma. Potrebbero essere presenti una o più funzioni di controllo del canale per un particolare gestore code.
2. L'implementazione degli MCA e delle funzioni di controllo del canale è altamente dipendente dalla piattaforma. Possono essere programmi o processi o thread, e possono essere una singola entità o molte che comprendono diverse parti indipendenti o collegate.
3. Tutti i componenti contrassegnati con un asterisco possono utilizzare MQI.

Parametri canale

Un MCA riceve i suoi parametri in uno dei seguenti modi:

- Se avviato da un comando, il nome del canale viene passato in un'area dati. L'MCA legge quindi la definizione del canale direttamente per ottenere i suoi attributi.
- Per i canali mittente e in alcuni casi server, l'MCA può essere avviato automaticamente dal trigger del gestore code. Il nome del canale viene richiamato dalla definizione del processo trigger, dove applicabile, e viene passato all'MCA. La restante elaborazione è la stessa descritta in precedenza. I canali server devono essere impostati per essere attivati solo se sono completi, ovvero, specificano un CONNAME a cui connettersi.
- Se avviato in remoto da un mittente, un server, un richiedente o una connessione client, il nome del canale viene trasmesso nei dati iniziali dall'agent del canale dei messaggi partner. MCA legge la definizione di canale direttamente per ottenere i suoi attributi.

Alcuni attributi non definiti nella definizione del canale sono anch'essi negoziabili:

Suddividi messaggi

Se un'estremità non supporta i messaggi suddivisi, i messaggi suddivisi non vengono inviati.

Capacità di conversione

Se un'estremità non può eseguire la conversione della codepage o la conversione della codifica numerica necessarie, l'altra estremità deve gestirla. Se nessuna delle due estremità lo supporta, quando necessario, il canale non può essere avviato.

Supporto elenco di distribuzione

Se un'estremità non supporta gli elenchi di distribuzione, l'MCA partner imposta un indicatore nella propria coda di trasmissione in modo che sappia intercettare i messaggi destinati a più destinazioni.

Stato del canale e numeri di sequenza

I programmi dell'agent del canale dei messaggi conservano i record del numero di sequenza corrente e del numero dell'unità di lavoro logica per ciascun canale e dello stato generale del canale. Alcune piattaforme consentono di visualizzare queste informazioni di stato per facilitare il controllo dei canali.

Come inviare un messaggio a un altro gestore code


In questa sezione viene descritto il modo più semplice per inviare un messaggio tra i gestori code, inclusi i prerequisiti e le autorizzazioni richieste. Altri metodi possono essere utilizzati anche per l'invio di messaggi a un gestore code remoto.

Prima di inviare un messaggio da un gestore code a un altro, è necessario effettuare le seguenti operazioni:



1. Verificare che il protocollo di comunicazione scelto sia disponibile.
2. Avviare i gestori code.
3. Avviare gli iniziatori di canali.
4. Avviare i listener.

È inoltre necessario disporre dell'autorizzazione di sicurezza IBM MQ corretta per creare gli oggetti richiesti.

Per inviare messaggi da un gestore code a un altro:

- Definire i seguenti oggetti sul gestore code di origine:
 - Canale di trasmissione
 - Definizione di coda remota
 - Coda di iniziazione ( richiesto su z/OS, altrimenti facoltativo)
 - Coda di trasmissione
 - Coda di messaggi non recapitabili
- Definire i seguenti oggetti sul gestore code di destinazione:
 - Canale di ricezione
 - Coda di destinazione
 - Coda di messaggi non recapitabili

È possibile utilizzare diversi metodi per definire questi oggetti, a seconda della piattaforma IBM MQ :

- Su tutte le piattaforme, è possibile utilizzare i comandi script IBM MQ (MQSC) descritti in [Comandi MQSC](#) i comandi PCF (programmable command format) descritti in [Automazione delle attività di gestione in IBM MQ Explorer](#).
-  Su z/OS, è anche possibile utilizzare i riquadri Operazione e Controllo descritti in [Amministrazione IBM MQ for z/OS](#) .
-  Su IBM i, è anche possibile utilizzare l'interfaccia pannello.

Per ulteriori informazioni sulla creazione dei componenti per l'invio di messaggi a un altro gestore code, consultare i seguenti argomenti secondari:

Concetti correlati

[“Tecniche di accodamento distribuito IBM MQ” a pagina 206](#)

Gli argomenti secondari in questa sezione descrivono le tecniche che sono di uso durante la pianificazione dei canali. Questi argomenti secondari descrivono le tecniche per pianificare come collegare i gestori code e gestire il flusso di messaggi tra le applicazioni.

[“Introduzione alla gestione delle code distribuite” a pagina 226](#)

DQM (Distributed Queue Management) viene utilizzato per definire e controllare le comunicazioni tra gestori code.

[“Attivazione dei canali” a pagina 250](#)

IBM MQ fornisce una funzione per avviare automaticamente un'applicazione quando vengono soddisfatte determinate condizioni su una coda. Questa funzione viene chiamata attivazione.

[“Sicurezza dei messaggi” a pagina 248](#)

Oltre alle tipiche funzioni di ripristino di IBM MQ, la gestione della coda distribuita garantisce che i messaggi vengano consegnati correttamente utilizzando una procedura del punto di sincronizzazione coordinata tra le due estremità del canale dei messaggi. Se questa procedura rileva un errore, chiude il canale in modo da poter esaminare il problema e mantiene i messaggi in modo sicuro nella coda di trasmissione fino a quando il canale non viene riavviato.

Attività correlate

[“Creazione di gestori code su più piattaforme” a pagina 7](#)

Prima di poter utilizzare messaggi e code, è necessario creare e avviare almeno un gestore code e i relativi oggetti associati. Un gestore code gestisce le risorse associate ad esso, in particolare le code di sua proprietà. Fornisce servizi di accodamento alle applicazioni per chiamate e comandi MQI (Message queuing Interface) per creare, modificare, visualizzare ed eliminare oggetti IBM MQ.

[“Monitoraggio e controllo dei canali su AIX, Linux, and Windows” a pagina 258](#)

Per DQM è necessario creare, monitorare e controllare i canali per i gestori code remoti. È possibile controllare i canali utilizzando comandi, programmi, IBM MQ Explorer, file per le definizioni dei canali e un'area di memoria per le informazioni di sincronizzazione.

[“Monitoraggio e controllo dei canali su IBM i” a pagina 282](#)

Utilizzare i comandi e i pannelli DQM per creare, monitorare e controllare i canali dei gestori code remoti. Ogni gestore code ha un programma DQM per il controllo delle interconnessioni ai gestori code remoti compatibili.

[“Configurazione delle connessioni tra client e server” a pagina 16](#)

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

[“Configurazione di un cluster di gestore code” a pagina 303](#)

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. È possibile definire componenti cluster e creare e gestire cluster.

[“Setting up communications with other queue managers on z/OS” a pagina 961](#)

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

Definizione dei canali

Per inviare i messaggi da un gestore code a un altro, è necessario definire due canali. È necessario definirne uno sul gestore code di origine e uno sul gestore code di destinazione.

Sul gestore code di origine

Definire un canale con un tipo di canale SENDER. È necessario specificare quanto segue:

- Il nome della coda di trasmissione da utilizzare (attributo XMITQ).
- Il nome della connessione del sistema partner (l'attributo CONNAME).

- Il nome del protocollo di comunicazione che si sta utilizzando (attributo TRPTYPE). Su IBM MQ for z/OS, il protocollo deve essere TCP o LU6.2. Su Multiplatforms, non occorre specificare il protocollo. È possibile lasciarlo per selezionare il valore dalla propria definizione di canale predefinita.

I dettagli di tutti gli attributi del canale sono forniti in [Attributi canale](#).

Sul gestore code di destinazione

Definire un canale con un tipo di canale RECEIVER e lo stesso nome del canale mittente.

Specificare il nome del protocollo di comunicazioni che si sta utilizzando (attributo TRPTYPE). Su IBM MQ for z/OS, il protocollo deve essere TCP o LU6.2. Su Multiplatforms, non occorre specificare il protocollo. È possibile lasciarlo per selezionare il valore dalla propria definizione di canale predefinita.

Le definizioni del canale ricevente possono essere generiche. Ciò significa che se si dispone di diversi gestori code che comunicano con lo stesso destinatario, i canali di invio possono tutti specificare lo stesso nome per il destinatario e una definizione di destinatario si applica a tutti.

Una volta definito il canale, è possibile verificarlo utilizzando il comando PING CHANNEL. Questo comando invia un messaggio speciale dal canale mittente al canale ricevente e verifica che venga restituito.

Nota: Il valore del parametro TRPTYPE viene ignorato dall'agent del canale dei messaggi che risponde. Ad esempio, un TRPTYPE di TCP sulla definizione del canale mittente inizia correttamente con un TRPTYPE LU62 sulla definizione del canale ricevente come partner.

Definizione delle code

Per inviare messaggi da un gestore code a un altro, è necessario definire fino a sei code. È necessario definire fino a quattro code sul gestore code di origine e fino a due code sul gestore code di destinazione.

Sul gestore code di origine

- Definizione di coda remota

In questa definizione, specificare quanto segue:

Nome gestore code remoto

Il nome del gestore code di destinazione.


Nome coda remota


Il nome della coda di destinazione sul gestore code di destinazione.

Nome coda di trasmissione


Il nome della coda di trasmissione. Non è necessario specificare questo nome della coda di trasmissione. In caso contrario, viene utilizzata una coda di trasmissione con lo stesso nome del gestore code di destinazione. Se non esiste, viene utilizzata la coda di trasmissione predefinita. Si consiglia di fornire alla coda di trasmissione lo stesso nome del gestore code di destinazione in modo che la coda venga trovata per impostazione predefinita.

- Definizione della coda di avvio

 È obbligatorio. È necessario utilizzare la coda di avvio denominata SYSTEM.CHANNEL.INITQ.

 Questo è facoltativo. Si consiglia di denominare la coda di iniziazione SYSTEM.CHANNEL.INITQ.

- Definizione della coda di trasmissione

Una coda locale con l'attributo USAGE impostato su XMITQ.  Se si utilizza l'interfaccia nativa IBM MQ for IBM i, l'attributo USAGE è *TMQ.

- Definizione coda di messaggi non instradabili

Definire una coda di messaggi non recapitabili in cui scrivere i messaggi non recapitati.

Sul gestore code di destinazione

- Definizione coda locale

La coda di destinazione. Il nome di questa coda deve essere uguale a quello specificato nel campo del nome della coda remota della definizione della coda remota sul gestore code di origine.

- Definizione coda di messaggi non instradabili

Definire una coda di messaggi non recapitabili in cui scrivere i messaggi non recapitati.

Concetti correlati

[“Creazione di una coda di trasmissione” a pagina 232](#)

Prima che un canale (diverso da un canale richiedente) possa essere avviato, la coda di trasmissione deve essere definita come descritto in questa sezione. La coda di trasmissione deve essere denominata nella definizione di canale.

[“Creazione di una coda di trasmissione su IBM i” a pagina 232](#)

È possibile creare una coda di trasmissione sulla piattaforma IBM i utilizzando il pannello Crea coda MQM.

Creazione di una coda di trasmissione

Prima che un canale (diverso da un canale richiedente) possa essere avviato, la coda di trasmissione deve essere definita come descritto in questa sezione. La coda di trasmissione deve essere denominata nella definizione di canale.

Definire una coda locale con l'attributo USAGE impostato su XMITQ per ogni canale di invio messaggi. Se si desidera utilizzare una coda di trasmissione specifica nelle definizioni della coda remota, creare una coda remota come mostrato.

Per creare una coda di trasmissione, utilizzare i comandi IBM MQ (MQSC), come mostrato nei seguenti esempi:

Crea esempio di coda di trasmissione

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

Crea esempio di coda remota

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +  
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

Considerare la possibilità di denominare la coda di trasmissione come nome del gestore code sul sistema remoto, come mostrato negli esempi.

Creazione di una coda di trasmissione su IBM i

È possibile creare una coda di trasmissione sulla piattaforma IBM i utilizzando il pannello Crea coda MQM.

È necessario definire una coda locale con l'attributo del campo Utilizzo impostato su *TMQ, per ciascun canale di messaggi di invio.

Se si desidera utilizzare le definizioni di coda remota, utilizzare lo stesso comando per creare una coda di tipo *RMT e Utilizzo di *NORMAL.

Per creare una coda di trasmissione, utilizzare il comando CRTMQ dalla riga comandi per visualizzare il pannello di creazione della prima coda; consultare [Figura 16 a pagina 233](#).


```

Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . .
Queue type . . . . . ____ *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . *DFT_____
-----

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
+

```

Figura 16. Crea una coda (1)

Immettere il nome della coda e specificare il tipo di coda che si desidera creare: Locale, Remota o Alias. Per una coda di trasmissione, specificare Locale (*LCL) su questo pannello e premere Invio. Viene visualizzata la seconda pagina del pannello Crea coda MQM; consultare [Figura 17 a pagina 233](#).

```

Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . . > HURS.2.HURS.PRIORIT
Queue type . . . . . > *LCL *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . *DFT
Replace . . . . . *NO *NO, *YES
Text 'description' . . . . .
Put enabled . . . . . *YES *SYSDFTQ, *NO, *YES
Default message priority . . . . 0 0-9, *SYSDFTQ
Default message persistence . . . *NO *SYSDFTQ, *NO, *YES
Process name . . . . .
Triggering enabled . . . . . *NO *SYSDFTQ, *NO, *YES
Get enabled . . . . . *YES *SYSDFTQ, *NO, *YES
Sharing enabled . . . . . *YES *SYSDFTQ, *NO, *YES

More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

Figura 17. Crea una coda (2)

Modificare i valori predefiniti visualizzati. Premere la pagina verso il basso per scorrere fino alla schermata successiva; vedere [Figura 18 a pagina 234](#).

```

Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Default share option . . . . . *YES      *SYSDFTQ, *NO, *YES
Message delivery sequence . . . *PTY      *SYSDFTQ, *PTY, *FIFO
Harden backout count . . . . . *NO      *SYSDFTQ, *NO, *YES
Trigger type . . . . . *FIRST      *SYSDFTQ, *FIRST, *ALL...
Trigger depth . . . . . 1          1-999999999, *SYSDFTQ
Trigger message priority . . . . 0          0-9, *SYSDFTQ
Trigger data . . . . . '          '
Retention interval . . . . . 999999999 0-999999999, *SYSDFTQ
Maximum queue depth . . . . . 5000     1-24000, *SYSDFTQ
Maximum message length . . . . . 4194304 0-4194304, *SYSDFTQ
Backout threshold . . . . . 0          0-999999999, *SYSDFTQ
Backout requeue queue . . . . . '          '
Initiation queue . . . . . '          '

More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

Figura 18. Crea una coda (3)

Immettere *TMQ, per la coda di trasmissione, nel campo Utilizzo di questo pannello e modificare i valori predefiniti mostrati negli altri campi.

```

Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Usage . . . . . *TMQ      *SYSDFTQ, *NORMAL, *TMQ
Queue depth high threshold . . . 80      0-100, *SYSDFTQ
Queue depth low threshold . . . 20      0-100, *SYSDFTQ
Queue full events enabled . . . *YES      *SYSDFTQ, *NO, *YES
Queue high events enabled . . . *YES      *SYSDFTQ, *NO, *YES
Queue low events enabled . . . *YES      *SYSDFTQ, *NO, *YES
Service interval . . . . . 999999999 0-999999999, *SYSDFTQ
Service interval events . . . . *NONE      *SYSDFTQ, *HIGH, *OK, *NONE
Distribution list support . . . *NO      *SYSDFTQ, *NO, *YES
Cluster Name . . . . . *SYSDFTQ
Cluster Name List . . . . . *SYSDFTQ
Default Binding . . . . . *SYSDFTQ *SYSDFTQ, *OPEN, *NOTFIXED

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

Figura 19. Crea una coda (4)

Quando si è soddisfatti che i campi contengono i dati corretti, premere Invio per creare la coda.

Avvio del canale

Quando si inseriscono i messaggi nella coda remota definita nel gestore code di origine, questi vengono memorizzati nella coda di trasmissione fino a quando il canale non viene avviato. Una volta avviato il canale, i messaggi vengono consegnati alla coda di destinazione sul gestore code remoto.

Avviare il canale sul gestore code di invio utilizzando il comando START CHANNEL. Quando si avvia il canale di invio, il canale di ricezione viene avviato automaticamente (dal listener) e i messaggi vengono inviati alla coda di destinazione. Entrambe le estremità del canale dei messaggi devono essere in esecuzione per poter trasferire i messaggi.

Poiché le due estremità del canale si trovano su gestori code differenti, è possibile che siano state definite con attributi differenti. Per risolvere eventuali differenze, esiste una negoziazione di dati iniziale tra le due estremità quando il canale viene avviato. In generale, le due estremità del canale operano con gli attributi che richiedono meno risorse. Ciò consente ai sistemi più grandi di contenere le risorse minori dei sistemi più piccoli all'altra estremità del canale dei messaggi.

L'MCA mittente suddivide i messaggi di grandi dimensioni prima di inviarli attraverso il canale. Vengono riassemblati sul gestore code remoto. Ciò non è evidente per l'utente.

Un MCA può trasferire messaggi utilizzando più thread. Questo processo, denominato *pipelining*, consente all'MCA di trasferire i messaggi in modo più efficiente, con meno stati di attesa. Pipelining migliora le prestazioni del canale.

Funzione di controllo canale

La funzione di controllo del canale consente di definire, monitorare e controllare i canali.

I comandi vengono emessi tramite pannelli, programmi o da una riga comandi alla funzione di controllo del canale. L'interfaccia del pannello visualizza anche lo stato del canale e i dati di definizione del canale. È possibile utilizzare Programmable Command Formats o quei comandi IBM MQ (MQSC) e i comandi di controllo descritti in [“Monitoraggio e controllo dei canali su AIX, Linux, and Windows”](#) a pagina 258.

I comandi rientrano nei seguenti gruppi:

- Amministrazione canale
- Controllo canale
- Monitoraggio stato canale

I comandi di gestione dei canali gestiscono le definizioni dei canali. Essi consentono di:

- Crea una definizione di canale
- Copia una definizione di canale
- Modifica di una definizione di canale
- Elimina una definizione di canale

I comandi di controllo del canale gestiscono il funzionamento dei canali. Essi consentono di:

- Avvia un canale
- Arresta un canale
- Risincronizza con il partner (in alcune implementazioni)
- Reimpostare i numeri di sequenza dei messaggi
- Risoluzione di un batch di messaggi in dubbio
- Ping; invia una comunicazione di verifica attraverso il canale

Il monitoraggio dei canali visualizza lo stato dei canali, ad esempio:

- Impostazioni canale correnti
- Se il canale è attivo o inattivo
- Se il canale è terminato in uno stato sincronizzato

Concetti correlati

[Dove trovare informazioni utili per la determinazione dei problemi](#)

Preparazione dei canali

Prima di tentare di avviare un canale messaggi o un canale MQI, è necessario preparare il canale. È necessario verificare che tutti gli attributi delle definizioni di canale locale e remoto sia corretti e compatibili.

[Attributi canale](#) descrive le definizioni e gli attributi del canale.

Anche se si impostano le definizioni di canale esplicite, le negoziazioni del canale eseguite all'avvio di un canale potrebbero sovrascrivere uno o l'altro dei valori definiti. Questo comportamento è normale, e non apparente per l'utente, ed è stato organizzato in questo modo in modo che le definizioni altrimenti incompatibili possano funzionare insieme.

Definizione automatica dei canali riceventi e di connessione server

In IBM MQ su Multiplatforms, se non esiste una definizione di canale appropriata, per un canale ricevente o di connessione server che ha la definizione automatica abilitata, viene creata automaticamente una definizione. La definizione viene creata utilizzando:

1. La definizione di canale modello appropriata, SYSTEM.AUTO.RECEIVERo SYSTEM.AUTO.SVRCONN. Le definizioni del canale modello per la definizione automatica sono le stesse dei valori predefiniti del sistema, SYSTEM.DEF.RECEIVERe SYSTEM.DEF.SVRCONN, ad eccezione del campo della descrizione, che è "Auto - definito da" seguito da 49 spazi. L'amministratore di sistema può scegliere di modificare qualsiasi parte delle definizioni di canale modello fornite.
2. Informazioni dal sistema partner. I valori del partner vengono utilizzati per il nome del canale e per il valore di wrap del numero di sequenza.
3. Un programma di uscita canale, che è possibile utilizzare per modificare i valori creati dalla definizione automatica. Vedere [Programma di uscita di definizione automatica del canale](#).

La descrizione viene quindi controllata per determinare se è stata modificata da un'uscita di definizione automatica o perché la definizione del modello è stata modificata. Se i primi 44 caratteri sono ancora "Definito automaticamente da" seguito da 29 spazi vuoti, il nome gestore code viene aggiunto. Se gli ultimi 20 caratteri sono ancora tutti vuoti, vengono aggiunti l'ora e la data locali.

Quando la definizione è stata creata e memorizzata, l'avvio del canale procede come se la definizione fosse sempre esistita. La dimensione batch, la dimensione di trasmissione e la dimensione del messaggio vengono negoziate con il partner.

Definizione di altri oggetti

Prima di poter avviare un canale di messaggi, entrambe le estremità devono essere definite (o abilitate per la definizione automatica) sui relativi gestori code. La coda di trasmissione che deve servire deve essere definita per il gestore code all'estremità di invio. Il collegamento di comunicazione deve essere definito e disponibile. Potrebbe essere necessario preparare altri oggetti IBM MQ , come le definizioni di code remote, le definizioni di alias del gestore code e le definizioni di alias della coda reply - to, per implementare gli scenari descritti in ["Configurazione dell'accodamento distribuito"](#) a pagina 205.

Per informazioni sulla definizione dei canali MQI, consultare ["Definizione di canali MQI"](#) a pagina 30.

Più canali di messaggi per coda di trasmissione

È possibile definire più di un canale per coda di trasmissione, ma solo uno di questi canali può essere attivo alla volta. Considerare questa opzione per il provisioning di instradamenti alternativi tra gestori code per il bilanciamento del traffico e collegare l'azione correttiva di errore. Una coda di trasmissione non può essere utilizzata da un altro canale se il canale precedente per utilizzarla è terminato lasciando un batch di messaggi in dubbio all'estremità di invio. Per ulteriori informazioni, consultare ["Gestione dei canali in dubbio"](#) a pagina 247.

Avvio di un canale

È possibile che un canale inizi a trasmettere i messaggi in uno dei quattro modi. Può essere:

- Avviato da un operatore (non da canali ricevente, ricevente del cluster o di connessione server).
- Attivato dalla coda di trasmissione. Questo metodo si applica solo ai canali mittente e ai canali server completi (quei canali che specificano un CONNAME). È necessario preparare gli oggetti necessari per attivare i canali.
- Avviato da un programma applicativo (non canali ricevente, ricevente cluster o di connessione server).
- Avviato in remoto dalla rete da un canale mittente, mittente cluster, richiedente, server o connessione client. I canali riceventi, riceventi del cluster e, possibilmente, i canali server e richiedenti vengono avviati in questo modo, così come i canali di connessione server. I canali stessi devono essere già avviati (cioè abilitati).

Nota: Poiché un canale è 'avviato', non trasmette necessariamente i messaggi. Invece, potrebbe essere 'abilitato' per avviare la trasmissione quando si verifica uno dei quattro eventi precedentemente descritti. L'abilitazione e la disabilitazione di un canale si ottiene utilizzando i comandi operatore START e STOP.

Stati del canale

Un canale può essere in uno dei tanti stati in qualsiasi momento. Alcuni stati hanno anche sottostati. Da un determinato stato un canale può spostarsi in altri stati.

La Figura 20 a pagina 237 mostra la gerarchia di tutti i possibili stati del canale e gli stati secondari che si applicano a ciascuno degli stati del canale.

Figura 21 a pagina 238 mostra i link tra gli stati del canale. Questi collegamenti si applicano a tutti i tipi di canali di messaggi e di canali di connessione server.

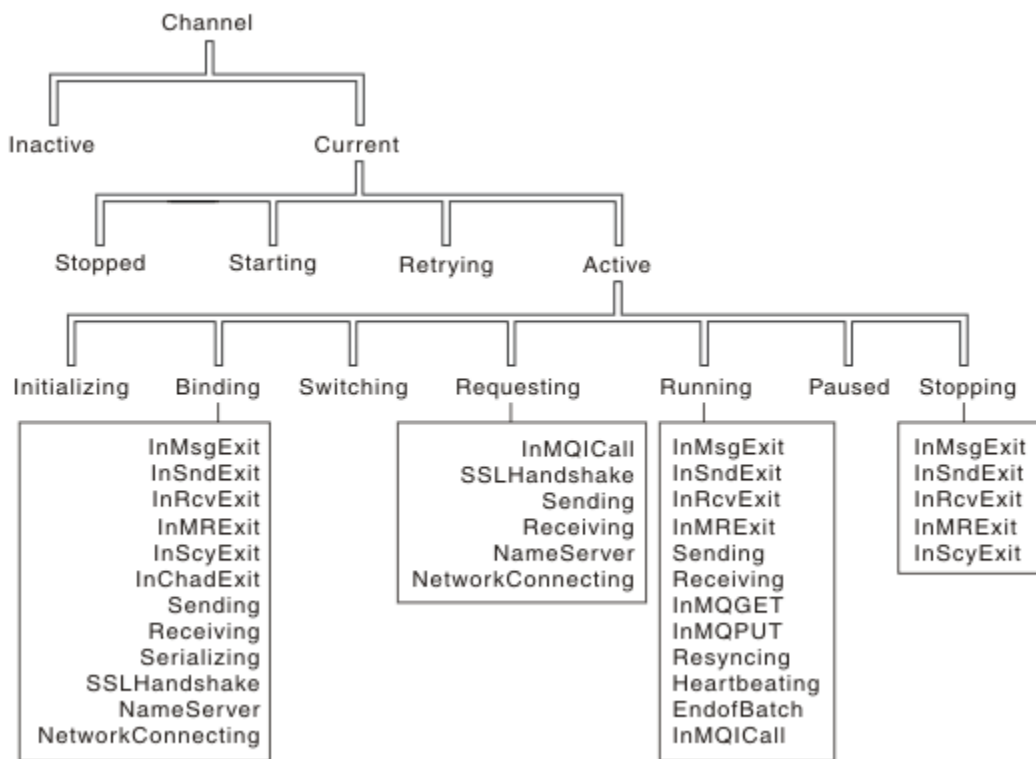


Figura 20. Stati e sottostati del canale

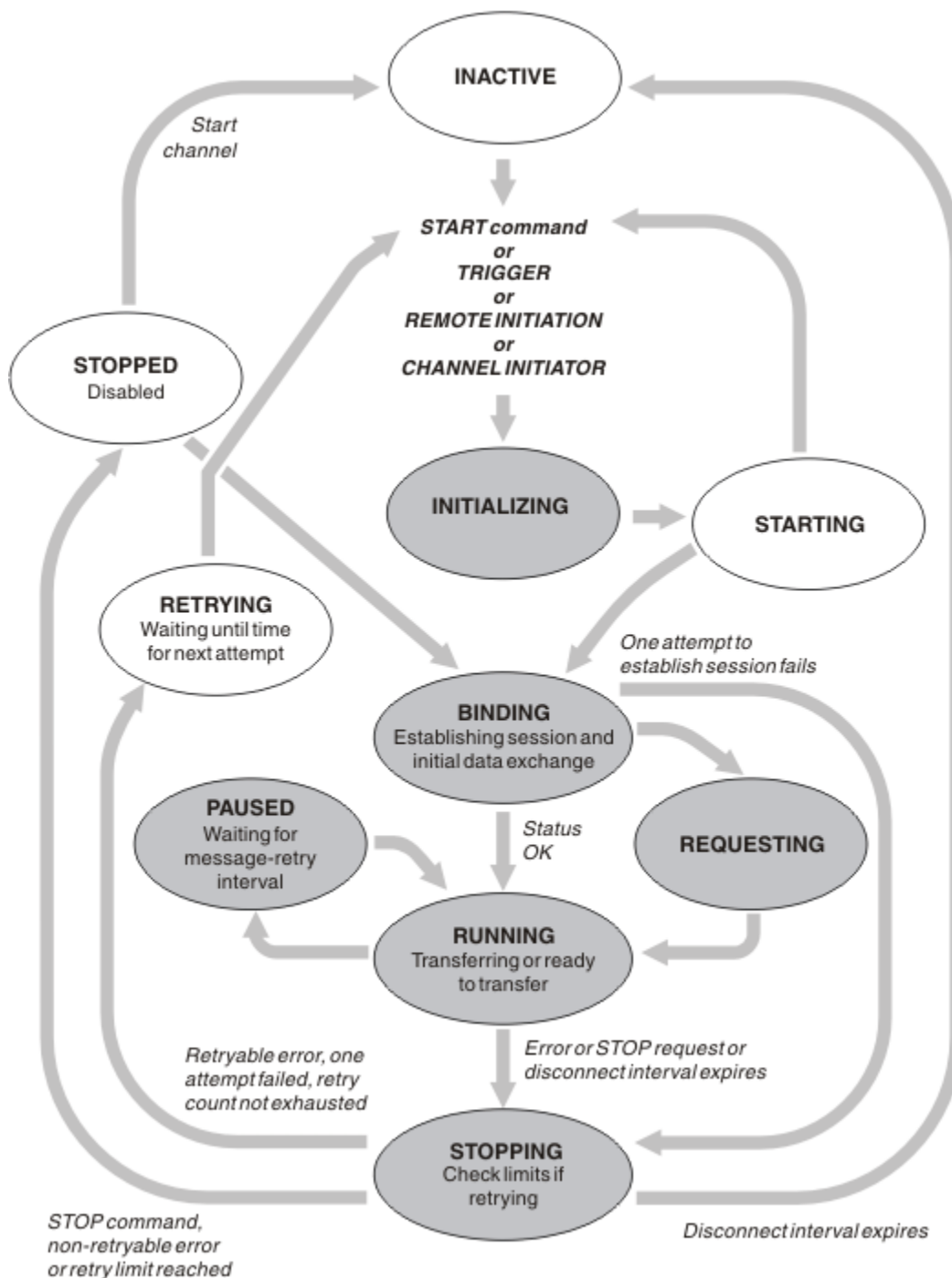


Figura 21. Flussi tra stati del canale

Corrente e attivo

Un canale è *corrente* se si trova in uno stato diverso da inattivo. Un canale corrente è *attivo* a meno che non si trovi nello stato NUOVO TENTATIVO, ARRESTATO o IN fase di avvio. Quando un canale è attivo, sta consumando una risorsa e un processo o un thread è in esecuzione. I sette possibili stati di un canale attivo (INITIALIZING, BINDING, SWITCHING, RICHIEDENTI, RUNNING, PAUSED o STOPPING) sono evidenziati in [Figura 21](#) a pagina 238.

Un canale attivo può anche visualizzare uno stato secondario che fornisce maggiori dettagli su cosa sta facendo esattamente il canale. Le sottostazioni per ciascuno stato sono mostrati in [Figura 20](#) a pagina 237.

Corrente e attivo

Il canale è "corrente" se si trova in uno stato diverso da inattivo. Un canale corrente è "attivo" a meno che non si trovi nello stato NUOVO TENTATIVO, ARRESTATO o IN fase di avvio.

Se un canale è "attivo" potrebbe anche mostrare un sottostato che fornisce maggiori dettagli su cosa sta facendo esattamente il canale.

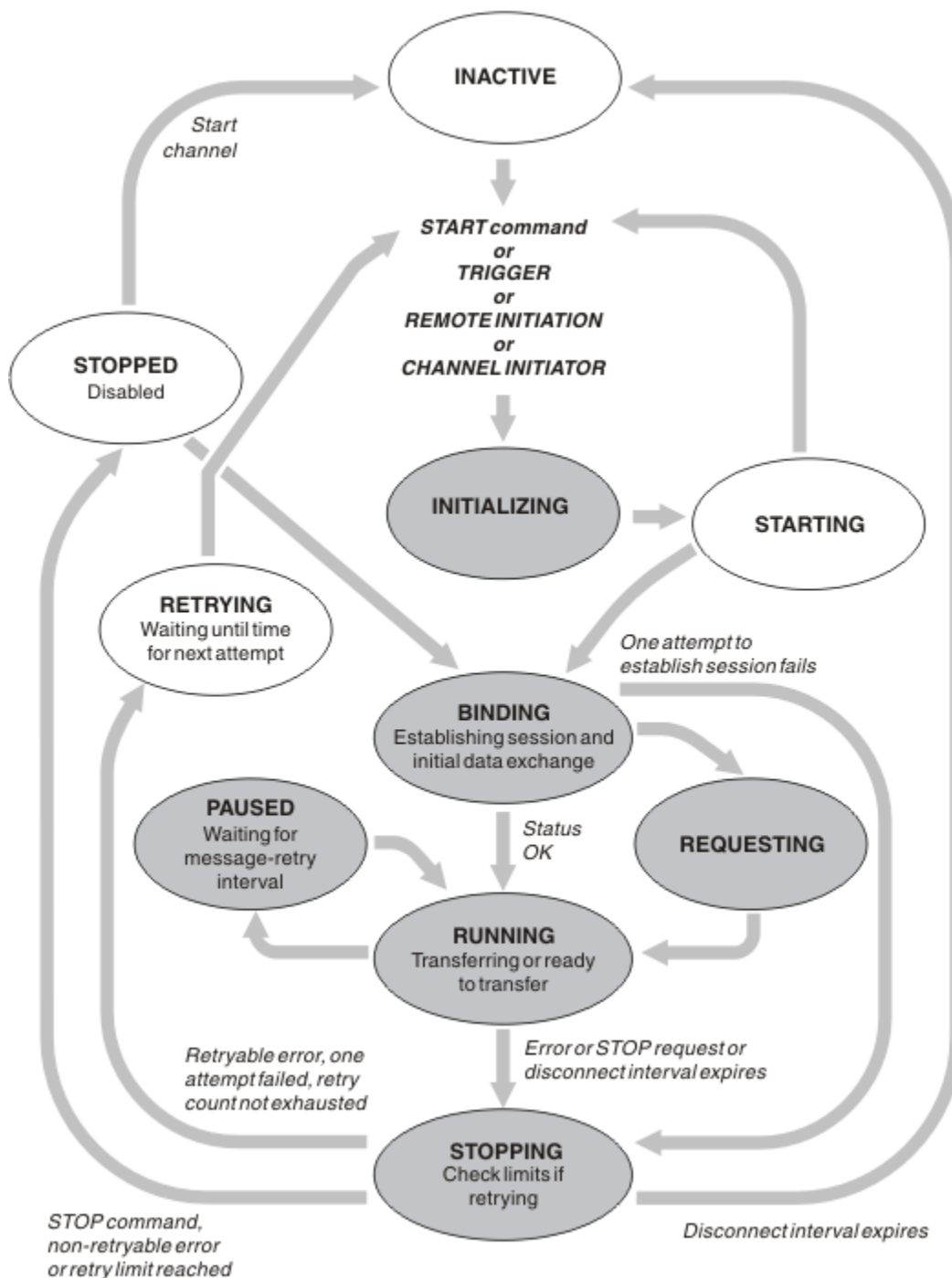


Figura 22. Flussi tra stati del canale

Nota:

1. Quando un canale si trova in uno dei sei stati evidenziati in Figura 22 a pagina 239 (INIZIALIZZAZIONE, LEGAMENTO, RICHIEDENTE, IN ESECUZIONE, IN PAUSA o SOSPENSIONE), sta consumando una risorsa e un processo o un thread è in esecuzione; il canale è *attivo*.

2. Quando un canale si trova nello stato ARRESTATO, la sessione potrebbe essere attiva perché lo stato successivo non è ancora noto.

Specifica del numero massimo di canali correnti

È possibile specificare il numero massimo di canali che possono essere correnti contemporaneamente. Questo numero è il numero di canali che hanno voci nella tabella di stato del canale, inclusi i canali che stanno ritentando e i canali che sono arrestati. Specificare quanto segue per la piattaforma:

- ▶ **z/OS** Utilizzare il comando ALTER QMGR MAXCHL .
- ▶ **IBM i** Modificare il file di inizializzazione del gestore code.
- ▶ **Linux** ▶ **AIX** Modificare il file di configurazione del gestore code.
- Utilizzare IBM MQ Explorer.

Per ulteriori informazioni sui valori impostati utilizzando il file di inizializzazione o di configurazione, consultare [Stanza del file di configurazione per l'accodamento distribuito](#). Per ulteriori informazioni su come specificare il numero massimo di canali, consultare i seguenti argomenti:

- ▶ **ALW** [Amministrazione di IBM MQ](#).
- ▶ **IBM i** [Amministrazione di IBM MQ for IBM i](#).
- ▶ **z/OS** [Amministrazione di IBM MQ for z/OS](#).

Nota:

1. I canali di connessione server sono inclusi in questo numero.
2. Un canale deve essere corrente prima di diventare attivo. Se un canale viene avviato, ma non può diventare corrente, l'avvio ha esito negativo.

Specifica del numero massimo di canali attivi

È anche possibile specificare il numero massimo di canali attivi per evitare che il sistema venga sovraccaricato da molti canali di avvio. Se si utilizza questo metodo, impostare l'attributo dell'intervallo di disconnessione su un valore basso per consentire l'avvio dei canali in attesa non appena terminano gli altri canali.

Ogni volta che un canale che tenta nuovamente di stabilire una connessione con il partner, deve diventare un canale attivo. Se il tentativo non riesce, rimane un canale corrente che non è attivo, fino a quando non è il momento per il successivo tentativo. Il numero di volte in cui un canale tenta nuovamente e la frequenza è determinata dagli attributi del numero di tentativi e dell'intervallo di tentativi. Esistono valori brevi e lunghi per entrambi questi attributi. Per ulteriori informazioni, consultare [Attributi del canale](#) .

Quando un canale deve diventare un canale attivo (perché è stato emesso un comando START, perché è stato attivato o perché è il momento di un nuovo tentativo), ma non è in grado di farlo perché il numero di canali attivi è già al valore massimo, il canale attende che uno degli slot attivi venga liberato da un'altra istanza del canale che cessa di essere attivo. Se, tuttavia, un canale viene avviato perché è stato avviato in remoto e non ci sono slot attivi disponibili per tale canale in quel momento, l'avvio remoto viene rifiutato.

Ogni volta che un canale, diverso da un canale richiedente, tenta di diventare attivo, passa allo stato STARTING. Questo stato si verifica anche se c'è uno slot attivo immediatamente disponibile, anche se è solo nello stato STARTING per un breve periodo di tempo. Tuttavia, se il canale deve attendere un alloggiamento attivo, è in stato AVVIO mentre è in attesa.

I canali del richiedente non passano allo stato STARTING. Se un canale richiedente non può essere avviato perché il numero di canali attivi è già al limite, il canale termina in modo anomalo.




Ogni volta che un canale, diverso da un canale richiedente, non è in grado di ottenere uno slot attivo e quindi ne attende uno, viene scritto un messaggio nel log ▶ **z/OS** o nella console z/OS , e viene generato un evento. Quando uno slot viene successivamente liberato e il canale è in grado di acquisirlo,

vengono generati un altro evento e un altro messaggio. Nessuno di questi eventi e messaggi viene generato se il canale è in grado di acquisire immediatamente uno slot.

Se si immette un comando STOP CHANNEL mentre il canale è in attesa di diventare attivo, il canale passa allo stato STOPPED. Viene generato un evento Arrestato dal canale.

I canali di connessione server sono compresi nel numero massimo di canali attivi.


Per ulteriori informazioni sulla specifica del numero massimo di canali attivi, consultare i seguenti argomenti:

-  Amministrazione di IBM MQ.
-  Amministrazione di IBM MQ for IBM i.
-  Amministrazione di IBM MQ for z/OS.


Errori canale


Gli errori sui canali causano l'arresto di ulteriori trasmissioni da parte del canale. Se il canale è un mittente o un server, passa allo stato RETRY perché è possibile che il problema si risolva da solo. Se non è possibile passare allo stato RETRY, il canale passa allo stato STOPPED.

Per i canali di invio, la coda di trasmissione associata è impostato su GET (DISABLED) e l'attivazione è disattivata. (Un comando STOP con STATUS (STOPPED) prende il lato che lo ha emesso nello stato STOPPED; solo la scadenza dell'intervallo di disconnessione o un comando STOP con STATUS (INACTIVE) lo rende terminato normalmente e diventa inattivo.) I canali che si trovano nello stato STOPPED necessitano dell'intervento dell'operatore prima di poter essere riavviati (consultare [“Riavvio dei canali arrestati”](#) a pagina 246).

Nota: Per sistemi  IBM i, AIX, Linux, and Windows , un iniziatore di canali deve essere in esecuzione per ritentare. Se l'iniziatore di canali non è disponibile, il canale diventa inattivo e deve essere riavviato manualmente. Se si sta utilizzando uno script per avviare il canale, assicurarsi che l'iniziatore del canale sia in esecuzione prima di provare ad eseguire lo script.

[Conteggio tentativi lunghi \(LONGRTY\)](#) descrive come funziona il nuovo tentativo. Se l'errore viene cancellato, il canale viene riavviato automaticamente e la coda di trasmissione viene riabilitata. Se il limite di tentativi viene raggiunto senza la cancellazione dell'errore, il canale passa allo stato ARRESTATO. Un canale arrestato deve essere riavviato manualmente dall'operatore. Se l'errore è ancora presente, non riprovare. Quando viene avviata correttamente, la coda di trasmissione viene riabilitata.

 Se l'iniziatore del canale si arresta mentre un canale si trova nello stato REENTAMENTO o ARRESTATO, lo stato del canale viene ricordato quando l'iniziatore del canale viene riavviato. Tuttavia, lo stato del canale per il tipo di canale SVRCONN viene reimpostato se l'iniziatore del canale si arresta mentre il canale è in stato ARRESTATO.

 Se il gestore code si arresta mentre un canale si trova nello stato REENTAMENTO o ARRESTATO, lo stato del canale viene ricordato quando il gestore code viene riavviato. Da IBM MQ 8.0 in poi, ciò si applica anche ai canali SVRCONN. In precedenza, lo stato del canale per il tipo di canale SVRCONN era stato reimpostato se l'iniziatore del canale era stato arrestato mentre il canale era in stato ARRESTATO.

Se un canale non è in grado di inserire un messaggio nella coda di destinazione perché tale coda è piena o non è consentita, il canale può ritentare l'operazione un certo numero di volte (specificato nell'attributo conteggio tentativi messaggi) in un intervallo di tempo (specificato nell'attributo intervallo tentativi messaggi). In alternativa, è possibile scrivere la propria uscita di nuovo tentativo di messaggio che determina le circostanze che causano un nuovo tentativo e il numero di tentativi effettuati. Il canale passa allo stato PAUSED durante l'attesa del completamento dell'intervallo di tentativi del messaggio.

Consultare [Attributi del canale](#) per informazioni relative agli attributi del canale e [Programmi di uscita del canale per i canali di messaggistica](#) per informazioni relative all'uscita del nuovo tentativo di messaggio.

Limiti del canale di connessione server

È possibile impostare i limiti del canale di connessione server per evitare che le applicazioni client esauriscano le risorse del canale del gestore code con il parametro **MAXINST** e per impedire che una singola applicazione client esaurisca la capacità del canale di connessione server con il parametro **MAXINSTC**.

Impostare **MAXINST** e **MAXINSTC** con il comando **DEFINE CHANNEL**.

Un numero massimo totale di canali può essere attivo in qualsiasi momento su un singolo gestore code. Il numero totale di istanze del canale di connessione server è incluso nel numero massimo di canali attivi.

Se non si specifica il numero massimo di istanze simultanee di un canale di connessione server che è possibile avviare, è possibile che una singola applicazione client, che si connette a un singolo canale di connessione server, esaurisca il numero massimo di canali attivi disponibili. Quando viene raggiunto il numero massimo di canali attivi, impedisce l'avvio di altri canali sul gestore code. Per evitare questa situazione, è necessario limitare il numero di istanze simultanee di un singolo canale di connessione server che è possibile avviare, indipendentemente dal client che le ha avviate.

Se il valore del limite viene ridotto al di sotto del numero di istanze attualmente in esecuzione del canale di connessione del server, anche a zero, i canali in esecuzione non vengono interessati. Le nuove istanze non possono essere avviate fino a quando non cessa l'esecuzione di istanze esistenti sufficienti in modo che il numero di istanze attualmente in esecuzione sia inferiore al valore del limite.

Inoltre, molti canali di connessione client diversi possono connettersi a un canale di connessione server individuale. Il limite sul numero di istanze simultanee di un singolo canale di connessione server che è possibile avviare, indipendentemente dal client che le ha avviate, impedisce a qualsiasi client di esaurire la capacità massima del canale attivo del gestore code. Se non si limita anche il numero di istanze simultanee di un singolo canale di connessione server che può essere avviato da un singolo client, è possibile che una singola applicazione client malfunzionante apra un numero di connessioni tale da esaurire la capacità del canale assegnata a un singolo canale di connessione server e quindi impedisca ad altri client che devono utilizzare il canale di connettersi ad esso. Per evitare questa situazione, è necessario limitare il numero di istanze simultanee di un singolo canale di connessione server che può essere avviato da un singolo client.

Se il valore del limite del singolo client viene ridotto al di sotto del numero di istanze del canale di connessione server attualmente in esecuzione da singoli client, anche a zero, i canali in esecuzione non vengono interessati. Tuttavia, le nuove istanze del canale di connessione server non possono essere avviate da un singolo client che supera il nuovo limite fino a quando non cessa l'esecuzione di un numero sufficiente di istanze esistenti da tale client, in modo che il numero di istanze attualmente in esecuzione sia inferiore al valore di questo parametro.

Riferimenti correlati

[Attributi e tipi di canale](#)

[Definire il canale](#)


Verifica che l'altra estremità del canale sia ancora disponibile

È possibile utilizzare l'intervallo di heartbeat, l'intervallo keep alive e il timeout di ricezione per verificare che l'altra estremità del canale sia disponibile.

Segnali di stato

È possibile utilizzare l'attributo del canale dell'intervallo heartbeat per specificare che i flussi devono essere passati dall'MCA di invio quando non vi sono messaggi nella coda di trasmissione, come descritto in [Intervallo heartbeat \(HBINT\)](#).

Keep alive

 Su z/OS, se si utilizza TCP/IP come protocollo di trasporto, è possibile anche specificare un valore per l'attributo del canale intervallo **Keepalive (KAINT)**. Si consiglia di assegnare all'intervallo **Keepalive** un valore più alto dell'intervallo di heartbeat e un valore più basso del valore di

disconnessione. È possibile utilizzare questo attributo per specificare un valore di timeout per ciascun canale, come descritto in [Intervallo keepalive \(KAINT\)](#).

Multi Su sistemi IBM i, AIX, Linux, and Windows , se si utilizza TCP come protocollo di trasporto, è possibile impostare `keepalive=yes`. Se si specifica questa opzione, TCP controlla periodicamente che l'altra estremità della connessione sia ancora disponibile. Non è così, il canale viene terminato. Questa opzione è descritta in [Intervallo keepalive \(KAINT\)](#).

Se si dispone di canali non affidabili che riportano errori TCP, l'utilizzo dell'opzione **Keepalive** indica che è più probabile che i canali vengano ripristinati.

È possibile specificare intervalli di tempo per controllare il comportamento dell'opzione **Keepalive** . Quando si modifica l'intervallo di tempo, vengono interessati solo i canali TCP/IP avviati dopo la modifica. Assicurarsi che il valore scelto per l'intervallo di tempo sia inferiore al valore dell'intervallo di disconnessione per il canale.

Per ulteriori informazioni sull'utilizzo dell'opzione **Keepalive** , consultare il parametro **KAINT** nel comando **DEFINE CHANNEL** .

Timeout ricezione

Se si utilizza TCP come protocollo di trasporto, anche l'estremità di ricezione di una connessione del canale non MQI inattiva viene chiusa se non vengono ricevuti dati per un periodo di tempo. Questo periodo, il valore di *timeout di ricezione* , viene determinato in base al valore **HBINT** (intervallo heartbeat).

In IBM MQ per sistemi IBM i, AIX, Linux, and Windows , il *timeout di ricezione* è impostato come segue:

1. Per un numero iniziale di flussi, prima di qualsiasi negoziazione, il valore *timeout di ricezione* è il doppio del valore **HBINT** dalla definizione del canale.
2. Una volta che i canali negoziano un valore **HBINT** , se **HBINT** è impostato su meno di 60 secondi, il valore *timeout di ricezione* è impostato sul doppio di questo valore. Se **HBINT** è impostato su 60 secondi o più, il valore di *timeout di ricezione* è impostato su 60 secondi maggiore del valore di **HBINT**.

z/OS Su z/OS, il valore *timeout di ricezione* è impostato come segue:

1. Per un numero iniziale di flussi, prima di qualsiasi negoziazione, il valore *timeout di ricezione* è il doppio del valore **HBINT** dalla definizione del canale.
2. Se **RCVTIME** è impostato, il timeout è impostato su uno dei seguenti valori, a seconda del parametro **RCVTYPE** , e soggetto a qualsiasi limite imposto da **RCVMTN** se si applica:
 - La **HBINT** negoziata moltiplicata per una costante
 - L'**HBINT** negoziato più un numero costante di secondi
 - Un numero costante di secondi

RCVMTN non si applica quando è configurato **RCVTYPE (EQUAL)** . Se si utilizza un valore costante di **RCVTIME** e si utilizza un intervallo di heartbeat, non specificare un valore **RCVTIME** inferiore all'intervallo di heartbeat. Per dettagli sugli attributi **RCVTIME**, **RCVMTN** e **RCVTYPE** , consultare il comando **ALTER QMGR** .

Nota:

1. Se uno dei valori è zero, non si verifica alcun timeout.
2. Per le connessioni che non supportano gli heartbeat, il valore **HBINT** viene negoziato a zero nel passo 2 e quindi non si verifica alcun timeout, pertanto è necessario utilizzare TCP/IP `KEEPALIVE`.
3. Per le connessioni client che utilizzano conversazioni condivise, gli heartbeat possono fluire attraverso il canale (da entrambe le estremità) tutto il tempo, non solo quando un `MQGET` è in sospeso.
4. Per le connessioni client in cui le conversazioni di condivisione non sono in uso, gli heartbeat vengono trasferiti dal server solo quando il client emette una chiamata `MQGET` con attesa. Pertanto, non si consiglia di impostare l'intervallo heartbeat troppo piccolo per i canali client. Ad esempio, se l'heartbeat è impostato su 10 secondi, una chiamata `MQCMIT` non riesce (con

MQRC_CONNECTION_BROKEN) se il commit impiega più di 20 secondi perché non sono stati trasmessi dati durante questo periodo di tempo. Ciò può accadere con grandi unità di lavoro. Tuttavia, non si verifica se vengono scelti i valori appropriati per l'intervallo di heartbeat poiché solo MQGET con attesa richiede periodi di tempo significativi.

Se **SHARECNV** è diverso da zero, il client utilizza una connessione full duplex, il che significa che il client può (e fa) heartbeat durante tutte le chiamate MQI

5. L'annullamento della connessione dopo il doppio dell'intervallo heartbeat è valido perché un flusso di dati o heartbeat è previsto almeno ad ogni intervallo heartbeat. L'impostazione dell'intervallo heartbeat troppo piccolo, tuttavia, può causare problemi, soprattutto se si utilizzano le uscite del canale. Ad esempio, se il valore di **HBINT** è un secondo e viene utilizzata un'uscita di invio o ricezione, l'estremità di ricezione attende solo 2 secondi prima di annullare il canale. Se l'MCA sta eseguendo un'attività come la codifica del messaggio, questo valore potrebbe essere troppo breve.

Impostazioni suggerite

IBM MQ for z/OS

Come punto di partenza iniziale, è possibile utilizzare:

```
/cpl ALTER QMGR TCPKEEP(YES) RCVTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

dove cpl è il prefisso del comando per il sottosistema del gestore code.

Per ulteriori informazioni sui vari parametri, vedere [ALTER QMGR](#) e [IBM MQ disponibilità di rete](#) .

Se l'indirizzo IP del mittente potrebbe essere tradotto in più di un indirizzo, potrebbe essere necessario impostare **ADOPTCHK** su QMNAME anziché su ALL.

IBM MQ for Multiplatforms

In qm.ini, aggiungere le seguenti informazioni:

```
TCP:
KeepAlive=Yes
CHANNELS:
AdoptNewMCA=ALL
AdoptNewMCACheck=ALL
```

Per ulteriori informazioni, consultare [ALTER QMGR](#), Stanza del file di configurazione per l'accodamento distribuito e “Stanza dei Canali del file qm.ini” a pagina 126 .

Se l'indirizzo IP del mittente può essere tradotto in più di un indirizzo, potrebbe essere necessario impostare **AdoptNewMCACheck** su QMNAME anziché su **ALL**.

Adozione di un MCA

La funzione Adotta MCA consente a IBM MQ di annullare un canale ricevente e avviarne uno nuovo al suo posto.

Se un canale perde il contatto, il canale ricevente può essere lasciato in uno stato di 'ricezione comunicazioni'. Quando le comunicazioni vengono ristabilite, il canale mittente tenta di riconnettersi. Se il gestore code remoto rileva che il canale ricevente è già in esecuzione, non consente l'avvio di un'altra versione dello stesso canale ricevente. Questo problema richiede l'intervento dell'utente per risolvere il problema o l'utilizzo del keepalive del sistema.

La funzione di adozione MCA risolve automaticamente il problema. Consente a IBM MQ di annullare un canale ricevente e di avviarne uno nuovo al suo posto.

Attività correlate

[Amministrazione IBM MQ](#)

[Amministrazione IBM MQ for z/OS](#)

Arresto e disattivazione dei canali

È possibile arrestare e sospendere un canale prima della scadenza dell'intervallo di tempo di disconnessione.

I canali di messaggi sono progettati per essere connessioni di lunga durata tra gestori code con terminazione ordinata controllata solo dall'attributo dell'intervallo di disconnessione del canale. Questo meccanismo funziona a meno che l'operatore non debba terminare il canale prima della scadenza dell'intervallo di tempo di disconnessione. Questa necessità può verificarsi nelle situazioni seguenti:

- Disattivazione del sistema
- Conservazione delle risorse
- Azione unilaterale a un'estremità di un canale

In questo caso, è possibile arrestare il canale. È possibile eseguire questa operazione utilizzando:

- comando STOP CHANNEL MQSC
- comando Arresta canale PCF
- IBM MQ Explorer

-   altri meccanismi specifici della piattaforma, come segue:

 **z/OS**

Per z/OS:

Il pannello Arresta un canale

 **IBM i**

Per IBM i:

Il comando CL ENDMQMCHL o l'opzione END sul pannello WRKMQMCHL

Esistono tre opzioni per arrestare i canali utilizzando questi comandi:

QUIESCE

L'opzione QUIESCE tenta di terminare il batch corrente di messaggi prima di arrestare il canale.

Forza

L'opzione FORCE tenta di arrestare il canale immediatamente e potrebbe richiedere la risincronizzazione del canale quando viene riavviato poiché il canale potrebbe essere lasciato in dubbio.

 **z/OS**

Su IBM MQ for z/OS, FORCE interrompe la riassegnazione dei messaggi in corso, lasciando i messaggi BIND_NOT_FIXED parzialmente riassegnati o non in ordine.

TERMINATE

L'opzione TERMINATE tenta di arrestare immediatamente il canale e termina il thread o il processo del canale.

 **z/OS**

Su IBM MQ for z/OS, TERMINATE interrompe la riassegnazione dei messaggi in corso, lasciando i messaggi BIND_NOT_FIXED parzialmente riassegnati o non in ordine.

Tutte queste opzioni lasciano il canale nello stato ARRESTATO, richiedendo l'intervento dell'operatore per riavviarlo.

L'arresto del canale all'estremità di invio è effettivo ma richiede l'intervento dell'operatore per il riavvio. All'estremità di ricezione del canale, le cose sono molto più difficili perché l'MCA è in attesa di dati dal lato di invio e non c'è modo di avviare una terminazione *ordinata* del canale dal lato di ricezione; il comando di arresto è in sospenso fino a quando l'MCA non ritorna dall'attesa dei dati.

Di conseguenza ci sono tre modi consigliati di utilizzare i canali, a seconda delle caratteristiche operative richieste:

- Se vuoi che i tuoi canali siano di lunga durata, tieni presente che ci può essere una terminazione ordinata solo dall'estremità di invio. Quando i canali vengono interrotti, ossia arrestati, è richiesto l'intervento dell'operatore (un comando START CHANNEL) per riavviarli.

- Se si desidera che i canali siano attivi solo quando vi sono messaggi da trasmettere, impostare l'intervallo di disconnessione su un valore abbastanza basso. L'impostazione predefinita è alta e quindi non è consigliata per i canali in cui è richiesto questo livello di controllo. Poiché è difficile interrompere il canale ricevente, l'opzione più economica è quella di disconnettere e riconnettere automaticamente il canale in base alle esigenze del workload. Per la maggior parte dei canali, l'impostazione appropriata dell'intervallo di disconnessione può essere stabilita in modo euristico.
- È possibile utilizzare l'attributo heartbeat - interval per fare in modo che l'MCA mittente invii un flusso heartbeat all'MCA ricevente durante i periodi in cui non ha messaggi da inviare. Questa azione rilascia l'MCA ricevente dal relativo stato di attesa e le fornisce l'opportunità di sospendere il canale senza attendere la scadenza dell'intervallo di disconnessione. Fornire all'intervallo di heartbeat un valore inferiore a quello dell'intervallo di disconnessione.

Nota:


1. Si consiglia di impostare l'intervallo di disconnessione su un valore basso o di utilizzare gli heartbeat per i canali del server. Questo valore basso è per consentire il caso in cui il canale richiedente termina in modo anomalo (ad esempio, perché il canale è stato annullato) quando non ci sono messaggi per il canale server da inviare. Se l'intervallo di disconnessione è impostato su un valore elevato e gli heartbeat non sono in uso, il server non rileva che il richiedente è terminato (operazione che eseguirà solo la volta successiva che tenterà di inviare un messaggio al richiedente). Mentre il server è ancora in esecuzione, mantiene la coda di trasmissione aperta per l'input esclusivo al fine di ottenere ulteriori messaggi che arrivano sulla coda. Se si tenta di riavviare il canale dal richiedente, la richiesta di avvio riceve un errore poiché il server ha ancora la coda di trasmissione aperta per l'input esclusivo. È necessario arrestare il canale del server e riavviare di nuovo il canale dal richiedente.


Riavvio dei canali arrestati

Quando un canale passa allo stato ARRESTATO, è necessario riavviare manualmente il canale.



Informazioni su questa attività

Per i canali mittente o server, quando il canale è entrato nello stato STOPPED, la coda di trasmissione associata è stata impostata su GET (DISABLED) e il trigger è stato disattivato. Quando viene ricevuta la richiesta di avvio, questi attributi vengono reimpostati automaticamente.

 Se l'iniziatore del canale si arresta mentre un canale si trova nello stato REENTAMENTO o ARRESTATO, lo stato del canale viene ricordato quando l'iniziatore del canale viene riavviato. Tuttavia, lo stato del canale per il tipo di canale SVRCONN viene reimpostato se l'iniziatore del canale si arresta mentre il canale è in stato ARRESTATO.

 Se il gestore code si arresta mentre un canale si trova nello stato REENTAMENTO o ARRESTATO, lo stato del canale viene ricordato quando il gestore code viene riavviato. Da IBM MQ 8.0 in poi, ciò si applica anche ai canali SVRCONN. In precedenza, lo stato del canale per il tipo di canale SVRCONN era stato reimpostato se l'iniziatore del canale era stato arrestato mentre il canale era in stato ARRESTATO.

Procedura

- Riavviare il canale in uno dei seguenti modi:
 - Utilizzando il [comando START CHANNEL MQSC](#).
 - Utilizzando il [comando Avvio canale PCF](#).
 - Utilizzando [IBM MQ Explorer](#)
 -  Su z/OS, utilizzando il [pannello Avvia un canale](#).
 -  Su IBM i, utilizzando il [comando CL STRMQMCHL](#) o l'opzione START sul [pannello WRMQMCHL](#).

Gestione dei canali in dubbio

Un canale in dubbio è un canale in dubbio con un canale remoto su cui sono stati inviati e ricevuti i messaggi.

Informazioni su questa attività

Notare la distinzione tra questo e un gestore code che è in dubbio su quali messaggi devono essere sottoposti a commit in una coda.

È possibile ridurre la possibilità che un canale venga messo in dubbio utilizzando il parametro del canale Batch Heartbeat (**BATCHHB**). Quando viene specificato un valore per questo parametro, un canale mittente controlla che il canale remoto sia ancora attivo prima di intraprendere qualsiasi ulteriore azione. Se non viene ricevuta alcuna risposta, il canale ricevente viene considerato non più attivo. I messaggi possono essere sottoposti a rollback e reinstradati e il canale mittente non viene messo in dubbio. Ciò riduce il tempo durante il quale il canale potrebbe essere messo in dubbio rispetto al periodo tra il canale mittente che verifica che il canale ricevente sia ancora attivo e che il canale ricevente abbia ricevuto i messaggi inviati. Consultare [Attributi del canale](#) per ulteriori informazioni sul parametro heartbeat batch.

I problemi del canale in dubbio vengono generalmente risolti automaticamente. Anche quando la comunicazione viene persa e un canale viene messo in dubbio con un batch di messaggi al mittente con stato di ricezione sconosciuto, la situazione viene risolta quando la comunicazione viene ristabilita. Il numero di sequenza e i record LUWID vengono conservati per questo scopo. Il canale è in dubbio fino a quando non sono state scambiate le informazioni LUWID e solo un batch di messaggi può essere in dubbio per il canale.

È possibile, quando necessario, risincronizzare manualmente il canale. Il termine manuale include l'utilizzo di operatori o programmi che contengono comandi di gestione del sistema IBM MQ. Il processo di risincronizzazione manuale funziona come segue. Questa descrizione utilizza i comandi MQSC, ma è anche possibile utilizzare gli equivalenti PCF.

Procedura

1. Utilizzare il comando **DISPLAY CHSTATUS** per trovare l'ultima LUWID (logical unit of work ID) con commit per ciascun lato del canale.

Effettuare questa operazione utilizzando i comandi riportati di seguito:

- Per il lato in dubbio del canale:

```
DISPLAY CHSTATUS(name) SAVED CURLUWID
```

È possibile utilizzare i parametri **CONNAME** e **XMITQ** per identificare ulteriormente il canale.

- Per il lato ricevente del canale:

```
DISPLAY CHSTATUS( name ) SAVED LSTLUWID
```

È possibile utilizzare il parametro **CONNAME** per identificare ulteriormente il canale.

Nota: I comandi sono diversi perché solo il lato mittente del canale può essere in dubbio. Il lato ricevente non è mai in dubbio.

 Su IBM i, il comando **DISPLAY CHSTATUS** può essere eseguito da un file utilizzando il comando **STRMQMMQSC** o il comando CL Work with MQM Channel Status, **WRKMQMCHST**.

2. Se i due LUWID sono gli stessi, utilizzare il comando **RESOLVE CHANNEL** per eseguire il commit dei messaggi in dubbio.


Se i due LUWID sono gli stessi, il lato ricevente ha eseguito il commit dell'unità di lavoro che il mittente considera in dubbio. Il lato mittente può ora rimuovere i messaggi in dubbio dalla coda di trasmissione e riabilitarli. Questa operazione viene eseguita con il seguente comando **RESOLVE CHANNEL** :

```
RESOLVE CHANNEL(name) ACTION(COMMIT)
```

3. Se i due LUWID sono differenti, utilizzare il comando **RESOLVE CHANNEL** per eseguire il backout dei messaggi in dubbio.

Se i due LUWID sono diversi, il lato ricevente non ha eseguito il commit dell'unità di lavoro che il mittente considera in dubbio. Il lato mittente deve conservare i messaggi in dubbio sulla coda di trasmissione e inviarli nuovamente. Questa operazione viene eseguita con il seguente comando **RESOLVE CHANNEL** :

```
RESOLVE CHANNEL( name ) ACTION(BACKOUT)
```

 Su IBM i, è possibile utilizzare il comando Resolve MQM Channel, **RSVMQMCHL**.

Risultati

Una volta completato questo processo, il canale non è più in dubbio. La coda di trasmissione può ora essere utilizzata da un altro canale, se necessario.

Riferimenti correlati

[DISPLAY CHSTATUS \(visualizzazione stato canale\)](#)

[RESOLVE CHANNEL \(chiedere a un canale di risolvere messaggi in dubbio\)](#)

Sicurezza dei messaggi

Oltre alle tipiche funzioni di ripristino di IBM MQ, la gestione della coda distribuita garantisce che i messaggi vengano consegnati correttamente utilizzando una procedura del punto di sincronizzazione coordinata tra le due estremità del canale dei messaggi. Se questa procedura rileva un errore, chiude il canale in modo da poter esaminare il problema e mantiene i messaggi in modo sicuro nella coda di trasmissione fino a quando il canale non viene riavviato.

La procedura del punto di sincronizzazione ha un ulteriore vantaggio in quanto tenta di recuperare una situazione *in dubbio* all'avvio del canale. (*In dubbio* è lo stato di un'unità di recupero per cui è stato richiesto un punto di sincronizzazione ma il risultato della richiesta non è ancora noto.) A questa funzione sono associate anche le due funzioni:

1. Risolvi con commit o backout
2. Reimposta il numero di sequenza

L'uso di queste funzioni si verifica solo in circostanze eccezionali perché il canale si ripristina automaticamente nella maggior parte dei casi.

Messaggi veloci, non persistenti

L'attributo del canale NPMSPEED (velocità dei messaggi non persistenti) può essere utilizzato per specificare che i messaggi non persistenti sul canale devono essere consegnati più rapidamente. Per ulteriori informazioni su questo attributo, consultare [Velocità messaggio non persistente \(NPMSPEED\)](#).

Se un canale termina mentre i messaggi veloci e non persistenti sono in transito, i messaggi potrebbero andare persi e spetta all'applicazione organizzarne il ripristino, se necessario.

Se il canale ricevente non può inserire il messaggio nella sua coda di destinazione, viene collocato nella coda di messaggi non recapitabili, se ne è stato definito uno. In caso contrario, il messaggio viene eliminato.

Nota: Se l'altra estremità del canale non supporta l'opzione, il canale viene eseguito alla velocità normale.

Messaggi non recapitati

Per informazioni su cosa accade quando un messaggio non può essere consegnato, consultare [“Cosa succede quando un messaggio non può essere consegnato?”](#) a pagina 249.

Cosa succede quando un messaggio non può essere consegnato?

Quando un messaggio non può essere consegnato, l'MCA può elaborarlo in diversi modi. Può riprovare, può tornare al mittente o può inserirlo nella coda di messaggi non recapitabili.

Figura 23 a pagina 249 mostra l'elaborazione che si verifica quando un MCA non è in grado di inserire un messaggio nella coda di destinazione. (Le opzioni mostrate non si applicano a tutte le piattaforme.)

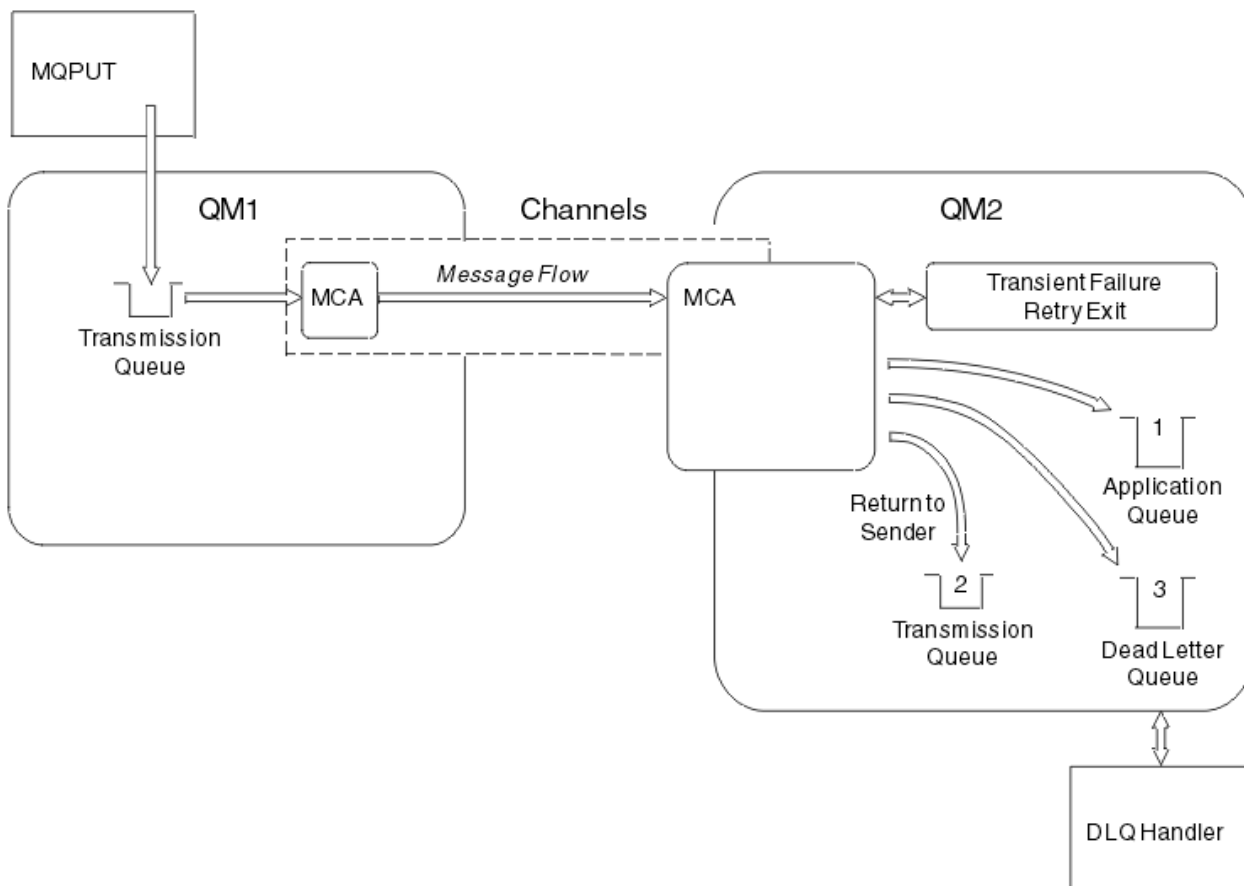


Figura 23. Cosa succede quando un messaggio non può essere consegnato

Come mostrato nella figura, l'MCA può eseguire diverse operazioni con un messaggio che non è in grado di consegnare. L'azione intrapresa è determinata dalle opzioni specificate quando il canale è definito e dalle opzioni del prospetto MQPUT per il messaggio.

1. tentativo messaggi

Se l'MCA non è in grado di inserire un messaggio nella coda di destinazione per un motivo che potrebbe essere transitorio (ad esempio, perché la coda è piena), l'MCA può attendere e ritentare l'operazione in un secondo momento. È possibile determinare se l'MCA attende, per quanto tempo e quante volte tenta.

- È possibile specificare un intervallo e un tempo di tentativi del messaggio per gli errori MQPUT quando si definisce il canale. Se il messaggio non può essere inserito nella coda di destinazione perché la coda è piena o non è consentita per le immissioni, l'MCA tenta l'operazione il numero di volte specificato, nell'intervallo di tempo specificato.

- È possibile scrivere la propria uscita di messaggi - tentativi. L'uscita consente di specificare in quali condizioni si desidera che l'MCA tenti nuovamente l'operazione MQPUT o MQOPEN. Specificare il nome dell'uscita quando si definisce il canale.

2. ritorno al mittente



Se il tentativo del messaggio ha avuto esito negativo o è stato rilevato un tipo di errore diverso, l'MCA può inviare nuovamente il messaggio al mittente. Per abilitare il ritorno al mittente, è necessario specificare le seguenti opzioni nel descrittore del messaggio quando si inserisce il messaggio nella coda originale:

- Opzione di report MQRO_EXCEPTION_WITH_FULL_DATA
- Opzione di report MQRO_DISCARD_MSG
- Il nome della coda di risposta e del gestore code di risposta

Se l'MCA non è in grado di inserire il messaggio nella coda di destinazione, genera un report di eccezioni contenente il messaggio originale e lo inserisce in una coda di trasmissione da inviare alla coda di risposta specificata nel messaggio originale. (Se la coda di risposta si trova sullo stesso gestore code dell'MCA, il messaggio viene inserito direttamente in quella coda, non in una coda di trasmissione.)

3. Coda di messaggi non recapitabili

Se un messaggio non può essere consegnato o restituito, viene inserito nella coda di messaggi non recapitabili (DLQ). È possibile utilizzare il gestore DLQ per elaborare il messaggio. Questa elaborazione è descritta qui:

-  [Elaborazione di messaggi su una coda di messaggi non instradabili](#)
-  [Programma di utilità gestore code di messaggi non recapitabili \(CSQUDLQH\)](#)

Se la coda di messaggi non instradabili non è disponibile, l'MCA di invio lascia il messaggio sulla coda di trasmissione e il canale si arresta. Su un canale veloce, i messaggi non persistenti che non possono essere scritti in una coda di messaggi non recapitabili vengono persi.

Su IBM WebSphere MQ 7.0, se non è definita alcuna coda di messaggi non recapitabili locale, la coda remota non è disponibile o definita e non è presente alcuna coda di messaggi non recapitabili remota, il canale mittente va in RETRY e i messaggi vengono automaticamente sottoposti a rollback nella coda di trasmissione.

Riferimenti correlati



[Utilizza coda di messaggi non instradabili \(USEDLQ\)](#)

Attivazione dei canali

IBM MQ fornisce una funzione per avviare automaticamente un'applicazione quando vengono soddisfatte determinate condizioni su una coda. Questa funzione viene chiamata attivazione.

Questa spiegazione è intesa come una panoramica dei concetti di attivazione. Per una descrizione completa, consultare [Avvio delle applicazioni IBM MQ utilizzando i trigger](#).

Per informazioni specifiche sulla piattaforma, consultare quanto segue:

- Per AIX, Linux, and Windows, consultare [“Attivazione dei canali su AIX, Linux, and Windows.”](#) a pagina 252
-  Per IBM i, consultare [“Attivazione dei canali in IBM MQ for IBM i”](#) a pagina 252
-  Per z/OS, consultare [“Transmission queues and triggering channels”](#) a pagina 964

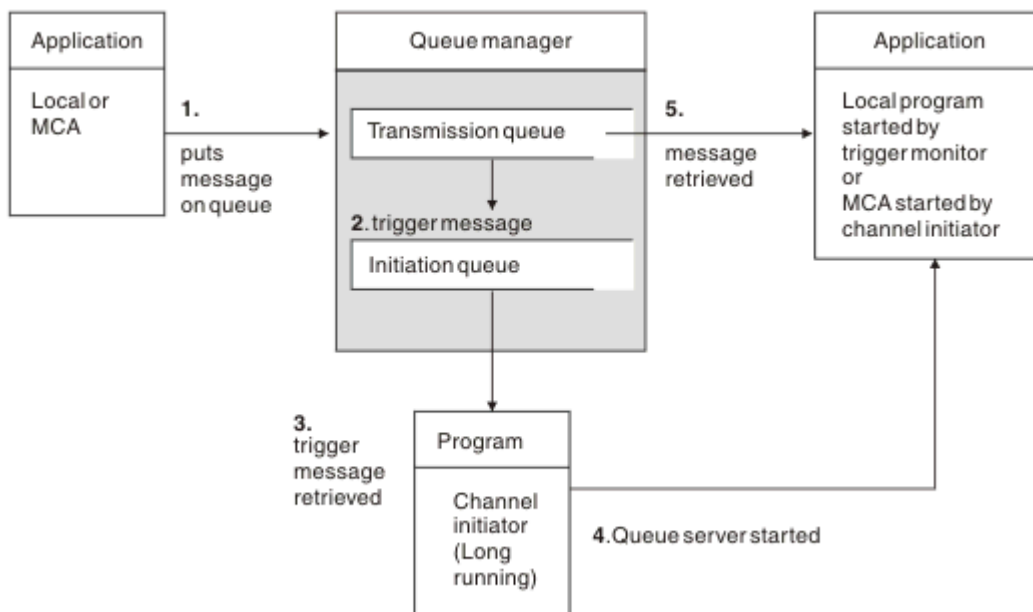


Figura 24. I concetti di attivazione

Gli oggetti richiesti per l'attivazione vengono mostrati in [Figura 24 a pagina 251](#). Mostra la seguente sequenza di eventi:

1. Il gestore code locale inserisce un messaggio da un'applicazione o da un MCA (message channel agent) nella coda di trasmissione.
2. Quando le condizioni di attivazione sono soddisfatte, il gestore code locale inserisce un messaggio di trigger nella coda di avvio.
3. Il programma iniziatore di canali di lunga durata controlla la coda di iniziazione e richiama i messaggi quando arrivano.
4. L'iniziatore del canale elabora i messaggi del trigger in base alle informazioni contenute in essi. Queste informazioni potrebbero includere il nome del canale, nel qual caso viene avviato l'MCA corrispondente.
5. L'applicazione locale o l'MCA, dopo essere stato attivato, richiama i messaggi dalla coda di trasmissione.

Per configurare questo scenario, è necessario:

- Creare la coda di trasmissione con il nome della coda di avvio (ovvero, SYSTEM.CHANNEL.INITQ) nell'attributo corrispondente.
- Assicurarsi che la coda di avvio (SYSTEM.CHANNEL.INITQ) esista.
- Verificare che il programma iniziatore di canali sia disponibile e in esecuzione. Il programma iniziatore di canali deve essere fornito con il nome della coda di iniziazione nel suo comando di avvio. **z/OS**
Su z/OS, il nome della coda di avvio è fisso, quindi non viene utilizzato sul comando di avvio.
- Facoltativamente, creare la definizione del processo per il trigger, se non esiste, e assicurarsi che nel campo *UserData* sia contenuto il nome del canale utilizzato. Invece di creare una definizione processo, è possibile specificare il nome del canale nell'attributo **TriggerData** della coda di trasmissione. IBM MQ per i sistemi **IBM i** IBM i, AIX, Linux, and Windows, consentono di specificare il nome del canale come vuoto, nel qual caso viene utilizzata la prima definizione di canale disponibile con questa coda di trasmissione.
- Accertarsi che la definizione della coda di trasmissione contenga il nome della definizione di processo per servirla (se applicabile), il nome della coda di avvio e le caratteristiche di attivazione che si ritengono

più adatte. L'attributo di controllo del trigger consente di abilitare o meno il trigger, in base alle necessità.

Nota:

1. Il programma iniziatore di canali agisce come un 'controllo trigger ' che controlla la coda di iniziazione utilizzata per avviare i canali.
2. Una coda di iniziazione e un processo trigger possono essere utilizzati per attivare un numero qualsiasi di canali.
3. È possibile definire qualsiasi numero di code di iniziazione e processi trigger.
4. Si consiglia un tipo di trigger FIRST, per evitare il riempimento del sistema con l'avvio del canale.

Attivazione dei canali su AIX, Linux, and Windows.



È possibile creare una definizione di processo in IBM MQ, definendo i processi da attivare. Utilizzare il comando MQSC DEFINE PROCESS per creare una definizione di processo che denomina il processo da attivare quando i messaggi arrivano su una coda di trasmissione. L'attributo USERDATA della definizione del processo contiene il nome del canale servito dalla coda di trasmissione.

Definire la coda locale (QM4), specificando che i messaggi trigger devono essere scritti nella coda di avvio (IQ) per attivare l'applicazione che avvia il canale (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

Definire l'applicazione (processo P1) da avviare:

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

In alternativa, per sistemi IBM MQ for UNIX, Linux e Windows , è possibile eliminare la necessità di una definizione di processo specificando il nome del canale nell'attributo TRIGDATA della coda di trasmissione.

Definire la coda locale (QM4). Specificare che i messaggi di trigger devono essere scritti nella coda di avvio predefinita SYSTEM.CHANNEL.INITQ, per attivare l'applicazione (processo P1) che avvia il canale (QM3.TO.QM4):

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)  
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

Se non si specifica un nome di canale, l'iniziatore del canale ricerca i file di definizione del canale finché non trova un canale associato alla coda di trasmissione denominata.

Attivazione dei canali in IBM MQ for IBM i



L'attivazione dei canali in IBM MQ for IBM i viene implementata con il processo dell'iniziatore di canali. Un processo iniziatore di canali per la coda di iniziazione SYSTEM.CHANNEL.INITQ viene avviato automaticamente con il gestore code, a meno che non sia disabilitato modificando l'attributo SCHINIT.

Impostare la coda di trasmissione per il canale, specificando SYSTEM.CHANNEL.INITQ come coda di iniziazione e abilitazione del trigger per la coda. L'iniziatore del canale avvia il primo canale disponibile che specifica questa coda di trasmissione.

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)
```

```
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)
USAGE(*TMQ)
```

Deprecated È possibile avviare manualmente fino a tre processi dell'iniziatore di canali con il comando STRMQMCHLI e specificare code di avvio differenti. È anche possibile specificare più di un canale in grado di elaborare la coda di trasmissione e scegliere quale canale avviare. Questa funzione è ancora fornita per essere compatibile con le release precedenti. Il suo utilizzo è obsoleto.

Nota: Solo un canale alla volta può elaborare una coda di trasmissione.

```
STRMQMCHLI QNAME(MYINITQ)
```

Impostare la coda di trasmissione per il canale, specificando TRGENBL (*YES) e, per scegliere quale canale tentare di avviare, specificare il nome del canale nel campo TRIGDATA. Ad esempio:

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(MYINITQ)
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

Concetti correlati

[“Avvio e arresto dell'iniziatore di canali” a pagina 253](#)

L'attivazione viene implementata utilizzando il processo dell'iniziatore di canali.

Attività correlate

[“Configurazione dell'accodamento distribuito” a pagina 205](#)

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

Riferimenti correlati

[Programmi di canale su AIX, Linux, and Windows](#)

 [Lavori di intercomunicazione su IBM i](#)

 [Stati del canale su IBM i](#)

Avvio e arresto dell'iniziatore di canali

L'attivazione viene implementata utilizzando il processo dell'iniziatore di canali.

Questo processo iniziatore di canali viene avviato con il comando MQSC START CHINIT. a meno che non si stia utilizzando la coda di avvio predefinita, specificare il nome della coda di avvio sul comando. Ad esempio, per utilizzare il comando START CHINIT per avviare il QI della coda per il gestore code predefinito, immettere:

```
START CHINIT INITQ(IQ)
```

Per impostazione predefinita, un iniziatore di canali viene avviato automaticamente utilizzando la coda di avvio predefinita, SYSTEM.CHANNEL.INITQ. Se si desidera avviare manualmente tutti gli iniziatori di canali, attenersi alla seguente procedura:

1. Creare e avviare il gestore code.
2. Modificare la proprietà SCHINIT del gestore code in MANUAL
3. Terminare e riavviare il gestore code

Nei sistemi IBM MQ for Multiplatforms , un iniziatore di canali viene avviato automaticamente. Il numero di iniziatori di canali che è possibile avviare è limitato. Il valore predefinito è massimo 3. È possibile modificare questa impostazione utilizzando MAXINITIATORS nel file qm.ini per sistemi AIX and Linux e nel registro per sistemi Windows .

Consultare [IBM MQ Comandi di controllo](#) per dettagli sul comando di esecuzione dell'iniziatore di canali **runmqchi** e sugli altri comandi di controllo.

Arresto dell'iniziatore di canali

L'iniziatore di canali predefinito viene avviato automaticamente quando si avvia un gestore code. Tutti gli iniziatori di canali vengono arrestati automaticamente quando un gestore code viene arrestato.

File di inizializzazione e di configurazione

La gestione dei dati di inizializzazione del canale dipende dalla piattaforma IBM MQ .

IBM MQ for z/OS



In IBM MQ for z/OS, le informazioni di inizializzazione e configurazione vengono specificate utilizzando il comando MQSC **ALTER QMGR** . Se si inseriscono i comandi **ALTER QMGR** nel dataset di input di inizializzazione CSQINP2 , questi vengono elaborati ogni volta che il gestore code viene avviato.

Per eseguire comandi MQSC come **START LISTENER** ogni volta che si avvia l'iniziatore di canali, inserirli nel dataset di input dell'inizializzazione CSQINPX e specificare l'istruzione DD facoltativa CSQINPX nella procedura dell'attività avviata dell'iniziatore di canali.

Per ulteriori informazioni su CSQINP2 e CSQINPX, consultare [Personalizzazione dei dataset di input di inizializzazione](#) e [ALTER QMGR](#).

IBM MQ for Multiplatforms



In IBM MQ for Multiplatforms, esistono file di configurazione per contenere le informazioni di configurazione di base sull'installazione di IBM MQ .

Ci sono due file di configurazione: uno si applica alla macchina, l'altro a un singolo gestore code.

IBM MQ file di configurazione

Questo file contiene informazioni relative a tutti i gestori code sul sistema IBM MQ . Il file si chiama `mqs.ini` . È descritto in ["File di configurazione IBM MQ , mqs.ini"](#) a pagina 94.

File di configurazione del gestore code

Questo file contiene informazioni di configurazione relative ad uno specifico gestore code. Il file è denominato `qm.ini` .

Viene creato durante la creazione del gestore code e può contenere le informazioni di configurazione relative a qualsiasi aspetto del gestore code. Le informazioni contenute nel file includono dettagli su come la configurazione del log differisce dal valore predefinito nel file di configurazione IBM MQ .

Il file di configurazione del gestore code si trova nella root della struttura di directory occupata dal gestore code. Ad esempio, per gli attributi **DefaultPath** , i file di configurazione del gestore code per un gestore code denominato QMNAME saranno:

Per sistemi AIX and Linux :

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Per sistemi Windows :

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```



Per IBM i:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

Ecco un estratto da un `qm.ini`. Specifica che il listener TCP/IP deve essere in ascolto sulla porta 2500, il numero massimo di canali correnti è 200 e il numero massimo di canali attivi è 100.

```
TCP:
Port=2500
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

È possibile specificare un intervallo di porte TCP/IP che devono essere utilizzate da un canale in uscita. Un metodo consiste nell'utilizzare il file `qm.ini` per specificare l'inizio e la fine di un intervallo di valori porta. Il seguente esempio mostra un file `qm.ini` che specifica un intervallo di canali:

```
TCP:
StrPort=2500
EndPort=3000
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

Se si specifica un valore per **StrPort** o **EndPort**, è necessario specificare un valore per entrambi. Il valore di **EndPort** deve essere sempre maggiore del valore di **StrPort**.

Il canale tenta di utilizzare ogni valore di porta nell'intervallo specificato. Quando la connessione riesce, il valore della porta è la porta utilizzata dal canale.

Per ulteriori informazioni sui file `qm.ini`, consultare [“File di configurazione del gestore code, qm.ini” a pagina 107](#).

Conversione dati per i messaggi

I messaggi IBM MQ potrebbero richiedere la conversione dei dati quando vengono inviati tra le code su gestori code differenti.

Un messaggio IBM MQ è composto da due parti:

- Informazioni di controllo in un descrittore di messaggi
- Dati applicazione

Una delle due parti potrebbe richiedere la conversione dei dati quando viene inviata tra le code su gestori code differenti. Per informazioni sulla conversione dei dati dell'applicazione, consultare [Conversione dei dati dell'applicazione](#).

Scrittura dei propri agent del canale dei messaggi

IBM MQ consente di scrivere i propri programmi MCA (message channel agent) o di installarne uno da un fornitore di software indipendente.

È possibile che si desideri scrivere i propri programmi MCA per rendere IBM MQ interoperativi sul proprio protocollo di comunicazione proprietario o per inviare messaggi su un protocollo non supportato da IBM MQ. (Non è possibile scrivere il proprio MCA per interagire con un MCA fornito da IBM MQ all'altra estremità.)

Se si decide di utilizzare un MCA non fornito da IBM MQ, è necessario considerare i seguenti punti.

Invio e ricezione di messaggi

È necessario scrivere un'applicazione di invio che riceva i messaggi da qualsiasi posizione in cui l'applicazione li inserisce, ad esempio da una coda di trasmissione e li invia su un protocollo con cui si desidera comunicare. È inoltre necessario scrivere un'applicazione ricevente che prenda i messaggi

da questo protocollo e li inserisca nelle code di destinazione. Le applicazioni di invio e di ricezione utilizzano le chiamate MQI (message queue interface), non interfacce speciali.

È necessario assicurarsi che i messaggi vengano consegnati solo una volta. Il coordinamento del punto di sincronizzazione può essere utilizzato per facilitare questa distribuzione.

Funzione di controllo canale

È necessario fornire le proprie funzioni di amministrazione per controllare i canali. Non è possibile utilizzare le funzioni di gestione del canale IBM MQ per la configurazione (ad esempio, il comando DEFINE CHANNEL) o il monitoraggio (ad esempio, DISPLAY CHSTATUS) dei canali.

File di inizializzazione

È necessario fornire il proprio file di inizializzazione, se necessario.

Conversione dati applicazione

È probabile che si desideri consentire la conversione dei dati per i messaggi inviati a un sistema differente. In tal caso, utilizzare l'opzione MQGMO_CONVERT nella chiamata MQGET quando si richiamano i messaggi da qualsiasi posizione in cui l'applicazione li inserisce, ad esempio la coda di trasmissione.

Uscite Utente

Considerare se sono necessarie uscite utente. In tal caso, è possibile utilizzare le stesse definizioni di interfaccia utilizzate da IBM MQ .

Triggering

Se l'applicazione inserisce i messaggi in una coda di trasmissione, è possibile impostare gli attributi della coda di trasmissione in modo che l'MCA di invio venga attivato quando i messaggi arrivano sulla coda.

Iniziatore di canali



È possibile che sia necessario fornire il proprio iniziatore di canali.


Altre cose da considerare per la gestione della coda distribuita

Altri argomenti da considerare quando si prepara IBM MQ per la gestione delle code distribuite. Questo argomento riguarda le code di messaggi non recapitati, le code in uso, le estensioni di sistema e i programmi di uscita utente e i canali e i listener in esecuzione come applicazioni attendibili.

Coda messaggi non recapitati

Per assicurarsi che i messaggi in arrivo sulla coda di messaggi non recapitati (nota anche come coda di messaggi non recapitabili o DLQ) vengano elaborati, creare un programma che possa essere attivato o eseguito a intervalli regolari per gestire tali messaggi.


  Un gestore DLQ viene fornito con IBM MQ su sistemi AIX and Linux ; per ulteriori informazioni, consultare [Il gestore DLQ di esempio, amqsdq](#).

 Per ulteriori informazioni su IBM MQ for IBM i, consultare [The IBM MQ for IBM i dead-letter queue handler](#).

Code in uso

Gli MCA per i canali riceventi possono mantenere aperte le code di destinazione anche quando i messaggi non vengono trasmessi. Ciò determina la visualizzazione delle code come "in uso".

Numero massimo di canali

 Su IBM MQ for IBM i è possibile specificare il numero massimo di canali consentiti nel proprio sistema e il numero massimo che può essere attivo contemporaneamente. Specificare questi numeri nel file `qm.ini` nell'indirizzario `QIBM/UserData/mqm/qmgrs/nome_gestore_code`. Consultare [Stanza del file di configurazione per l'accodamento distribuito](#).

Estensioni di sistema e programmi di uscita utente

Viene fornita una funzionalità nella definizione del canale per consentire l'esecuzione di programmi aggiuntivi in momenti definiti durante l'elaborazione dei messaggi. Questi programmi non vengono forniti con IBM MQ, ma possono essere forniti da ciascuna installazione in base ai requisiti locali.

Per poter essere eseguiti, questi programmi di uscita utente devono avere nomi predefiniti e devono essere disponibili durante la chiamata ai programmi del canale. I nomi dei programmi di uscita utente sono inclusi nelle definizioni del canale messaggi.

Esiste un'interfaccia di blocco di controllo definita per la consegna del controllo a questi programmi e per la gestione della restituzione del controllo da questi programmi.

Le posizioni precise in cui vengono richiamati questi programmi e i dettagli dei nomi e dei blocchi di controllo si trovano in [Programmi di uscita canale per i canali di messaggistica](#).

Esecuzione di canali e listener come applicazioni attendibili

Se le prestazioni sono una considerazione importante nell'ambiente e l'ambiente è stabile, è possibile eseguire i canali e i listener come attendibili, utilizzando il bind FASTPATH. Ci sono due fattori che influenzano se i canali e i listener vengono eseguiti come attendibili:

- La variabile di ambiente MQ_CONNECT_TYPE=FASTPATH o MQ_CONNECT_TYPE = STANDARD. È sensibile al maiuscolo / minuscolo. Se si specifica un valore non valido, viene ignorato.
- MQIBindType nella stanza Channels del file `qm.ini` o del file di registro. È possibile impostarlo su FASTPATH o STANDARD e non è sensibile al maiuscolo / minuscolo. L'impostazione predefinita è Standard.

È possibile utilizzare MQIBindType in associazione con la variabile di ambiente per ottenere l'effetto richiesto nel modo seguente:

MQIBindType	Variabile di ambiente	Risultato
STANDARD	NON DEFINITO	STANDARD
Percorso veloce	NON DEFINITO	Percorso veloce
STANDARD	STANDARD	STANDARD
Percorso veloce	STANDARD	STANDARD
STANDARD	Percorso veloce	STANDARD
Percorso veloce	Percorso veloce	Percorso veloce
STANDARD	CLIENTE	CLIENTE
Percorso veloce	CLIENTE	STANDARD
STANDARD	LOCALE	STANDARD
Percorso veloce	LOCALE	STANDARD

In sintesi, ci sono solo due modi per rendere realmente affidabili i canali e i listener:

1. Specificando MQIBindType= FASTPATH in `qm.ini` o nel registro e non specificando la variabile di ambiente.
2. Specificando MQIBindType= FASTPATH in `qm.ini` o nel registro e impostando la variabile di ambiente su FASTPATH.

Considerare l'esecuzione dei listener come attendibili, poiché i listener sono processi stabili. Considerare l'esecuzione dei canali come sicuri, a meno che non si stiano utilizzando uscite di canali instabili o il comando STOP CHANNEL MODE (TERMINATE).

Per DQM è necessario creare, monitorare e controllare i canali per i gestori code remoti. È possibile controllare i canali utilizzando comandi, programmi, IBM MQ Explorer, file per le definizioni dei canali e un'area di memoria per le informazioni di sincronizzazione.

Informazioni su questa attività

È possibile utilizzare i seguenti tipi di comandi per controllare i canali:

I comandi IBM MQ (MQSC)

È possibile utilizzare MQSC come singoli comandi in una sessione MQSC in sistemi AIX, Linux, and Windows . Per emettere comandi più complicati o multipli, è possibile creare MQSC in un file che si esegue dalla riga comandi. Per i dettagli, consultare [Comandi MQSC](#). Questa sezione fornisce alcuni semplici esempi di utilizzo di MQSC per l'accodamento distribuito.

I comandi del canale sono un sottoinsieme di MQSC (IBM MQ Commands). Utilizzare MQSC e i comandi di controllo per:

- Creare, copiare, visualizzare, modificare ed eliminare le definizioni di canale
- Avviare e arrestare i canali, eseguire il ping, ripristinare i numeri di sequenza dei canali e risolvere i messaggi in dubbio quando i collegamenti non possono essere ristabiliti
- Visualizza informazioni di stato sui canali

Comandi di controllo

È inoltre possibile immettere i *comandi di controllo* dalla riga comandi per alcune di tali funzioni. Per i dettagli, consultare [Amministrazione IBM MQ for Multiplatforms utilizzando i comandi di controllo](#).

Comandi del formato del comando programmabile

Per dettagli, consultare [Comandi PCF](#).

 Linux Windows

IBM MQ Explorer

Su sistemi Linux e Windows , è possibile utilizzare IBM MQ Explorer. Ciò fornisce un'interfaccia di gestione grafica per eseguire le attività di gestione come alternativa all'utilizzo dei comandi di controllo o dei comandi MQSC. Le definizioni di canale vengono conservate come oggetti del gestore code.

Ogni gestore code dispone di un componente DQM per il controllo delle interconnessioni ai gestori code remoti compatibili. Un'area di memoria contiene numeri di sequenza e identificatori *LUW (logical unit of work)* . Vengono utilizzati per scopi di sincronizzazione del canale.

Per un elenco delle funzioni disponibili durante l'impostazione e il controllo dei canali di messaggi, utilizzando i diversi tipi di comando, consultare [Tabella 22 a pagina 259](#).

Procedura

- [“Funzioni richieste per l'impostazione e il controllo dei canali” a pagina 259](#)
- [“Introduzione agli oggetti” a pagina 261](#)
- [“Impostazione della comunicazione su Windows” a pagina 268](#)
- [“Impostazione della comunicazione su AIX and Linux” a pagina 275](#)

Attività correlate

[“Monitoraggio e controllo dei canali su IBM i” a pagina 282](#)

Utilizzare i comandi e i pannelli DQM per creare, monitorare e controllare i canali dei gestori code remoti. Ogni gestore code ha un programma DQM per il controllo delle interconnessioni ai gestori code remoti compatibili.

Riferimenti correlati

 ALW

[Programmi di canale su AIX, Linux, and Windows](#)

 ALW

[Esempio di pianificazione del canale di messaggi per AIX, Linux, and Windows](#)

ALW Funzioni richieste per l'impostazione e il controllo dei canali

Potrebbe essere necessario un certo numero di funzioni IBM MQ per impostare e controllare i canali. Le funzioni del canale sono illustrate in questo argomento.

È possibile creare una definizione di canale utilizzando i valori predefiniti forniti da IBM MQ, specificando il nome del canale, il tipo di canale che si sta creando, il metodo di comunicazione da utilizzare, il nome coda di trasmissione e il nome connessione.

Il nome del canale deve essere lo stesso ad entrambe le estremità del canale e univoco nella rete. Tuttavia, è necessario limitare i caratteri utilizzati a quelli validi per i nomi oggetto IBM MQ .

Per altre funzioni correlate al canale, consultare i seguenti argomenti:

- [“Introduzione agli oggetti” a pagina 261](#)
- [“Creazione di oggetti associati” a pagina 262](#)
- [“Creazione di oggetti predefiniti” a pagina 262](#)
- [“Creazione di un canale” a pagina 262](#)
- [“Visualizzazione di un canale” a pagina 263](#)
- [“Visualizzazione dello stato del canale” a pagina 263](#)
- [“Controllo dei collegamenti mediante ping” a pagina 264](#)
- [“Avvio di un canale” a pagina 264](#)
- [“Arresto di un canale” a pagina 266](#)
- [“Ridenominazione di un canale” a pagina 266](#)
- [“Reimpostazione di un canale” a pagina 267](#)
- [“Risoluzione dei messaggi in dubbio su un canale” a pagina 267](#)



Tabella 22 a pagina 259 mostra l'elenco completo delle funzioni IBM MQ di cui potresti aver bisogno.

<i>Tabella 22. Funzioni richieste nei sistemi AIX, Linux, and Windows</i>			
Funzione	Comandi di controllo	MQSC	IBM MQ Equivalente di Explorer?
Funzioni del gestore code			
Modifica gestore code		ALTER DRG	Sì
Creare il gestore code	qmqm		Sì
Elimina gestore code	dlmqm		Sì
Visualizza gestore code		VISUALIZZA QMGR	Sì
Termina gestore code	qmfine		Sì
Ping gestore code		QMGR PING	No
Avvia gestore code	strmqm		Sì
Funzioni del server dei comandi			
Visualizza server dei comandi	vmqcsv		No
Termina server dei comandi	endmqcsv		No
Avvio server dei comandi	strmqcsv		No

Tabella 22. Funzioni richieste nei sistemi AIX, Linux, and Windows (Continua)

Funzione	Comandi di controllo	MQSC	IBM MQ Equivalente di Explorer?
Funzioni della coda			
Modifica coda		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE Consultare <u>Code ALTER.</u>	Sì
Cancella coda		<u>CANCELLA QLOCAL</u>	Sì
Creazione coda		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE Consultare <u>DEFINE queues.</u>	Sì
Elimina coda		ELIMINARE QALIAS ELIMINARE QLOCAL ELIMINARE QMODEL ELIMINARE QREMOTE Vedere code <u>DELETE.</u>	Sì
Visualizza coda		<u>VISUALIZZA CODA</u>	Sì
Funzioni di processo			
Modifica processo		<u>MODIFICA PROCESSO</u>	Sì
Crea processo		<u>DEFINE PROCESS</u>	Sì
Elimina processo		<u>Eliminazione processo</u>	Sì
Visualizza processo		<u>VISUALIZZA PROCESSO</u>	Sì
Funzioni di canale			
Modifica canale		<u>MODIFICA CANALE</u>	Sì
Crea canale		<u>Definire il canale</u>	Sì
Elimina canale		<u>Elimina canale</u>	Sì
Visualizza canale		<u>VISUALIZZA CANALE</u>	Sì
Visualizza stato canale		<u>VISUALIZZA CHSTATUS</u>	Sì
Fine canale		<u>Arresto canale</u>	Sì

Tabella 22. Funzioni richieste nei sistemi AIX, Linux, and Windows (Continua)

Funzione	Comandi di controllo	MQSC	IBM MQ Equivalente di Explorer?
Ping canale		Ping canale	Sì
Reimposta canale		Reimpostazione canale	Sì
Risoluzione canale		Risoluzione canale	Sì
Esegui canale	runmqchl	Avvio canale	Sì
 Esegui iniziatore di canali	runmqchi	INIZIO STRINGA	No
Esegui listener ¹	runmqlsr	Avvia listener	No
Fine listener	endmqlsr, solo sulle piattaforme seguenti: <ul style="list-style-type: none"> •  AIX •  Sistemi Windows 		No

Nota:

1. Un listener potrebbe essere avviato automaticamente all'avvio del gestore code.

 **Introduzione agli oggetti**

I canali devono essere definiti e i relativi oggetti associati devono esistere ed essere disponibili per l'utilizzo, prima che un canale possa essere avviato. Questa sezione mostra come.

Utilizzare i comandi IBM MQ (MQSC) o IBM MQ Explorer per:

1. Definire canali di messaggi e oggetti associati
2. Monitorare e controllare i canali dei messaggi

Gli oggetti associati che potrebbe essere necessario definire sono:

- Code di trasmissione
- Definizioni di coda remota
- Definizioni alias gestore code
- Definizioni alias coda di risposta
- Code locali di risposta
- Processi per l'attivazione (MCA)
- Definizioni di canali di messaggi

Il particolare collegamento di comunicazione per ciascun canale deve essere definito e disponibile prima che un canale possa essere eseguito. Per una descrizione della modalità di definizione dei collegamenti LU 6.2, TCP/IP, NetBIOS, SPX e DECnet, consultare la specifica guida alla comunicazione per l'installazione. Consultare anche [Informazioni di configurazione di esempio](#).

Per ulteriori informazioni sulla creazione e l'utilizzo degli oggetti, consultare i topic secondari riportati di seguito:

Creazione di oggetti associati

MQSC viene utilizzato per creare oggetti associati.

Utilizzare MQSC per creare la coda e gli oggetti alias: code di trasmissione, definizioni di code remote, definizioni di alias del gestore code, definizioni di alias della coda reply - to e code locali reply - to.

Creare anche le definizioni dei processi per l'attivazione (MCA) in modo simile.

Per un esempio che mostra come creare tutti gli oggetti richiesti, consultare [Esempio di pianificazione del canale dei messaggi per AIX, Linux, and Windows](#).

Creazione di oggetti predefiniti

Gli oggetti predefiniti vengono creati automaticamente quando viene creato un gestore code. Questi oggetti sono code, canali, una definizione di processo e code di gestione. Una volta creati gli oggetti predefiniti, è possibile sostituirli in qualsiasi momento eseguendo il comando `strmqm` con l'opzione `-c`.

Quando si utilizza il comando `crtmqm` per creare un gestore code, il comando avvia anche un programma per creare una serie di oggetti predefiniti.

1. Ogni oggetto predefinito viene creato a turno. Il programma conserva un conteggio del numero di oggetti definiti con esito positivo, del numero di oggetti esistenti e sostituiti e del numero di tentativi non riusciti.
2. Il programma visualizza i risultati e, se si sono verificati degli errori, indirizza l'utente al log degli errori appropriato per i dettagli.

Una volta terminata l'esecuzione del programma, è possibile utilizzare il comando `strmqm` per avviare il gestore code.

Consultare [Amministrazione IBM MQ for Multiplatforms utilizzando i comandi di controllo](#) per ulteriori informazioni sui comandi `crtmqm` e `strmqm`.

Modifica degli oggetti predefiniti

Quando si specifica l'opzione `-c`, il gestore code viene avviato temporaneamente mentre gli oggetti vengono creati e viene quindi nuovamente arrestato. L'emissione di `strmqm` con l'opzione `-c` aggiorna gli oggetti di sistema esistenti con i valori predefiniti (ad esempio, l'attributo `MCAUSER` di una definizione di canale è impostato su spazi vuoti). È necessario utilizzare nuovamente il comando `strmqm`, senza l'opzione `-c`, se si desidera avviare il gestore code.

Se si desidera modificare gli oggetti predefiniti, è possibile creare la propria versione del vecchio file `amqscoma.tst` e modificarla.

Creazione di un canale

Creare due definizioni di canale, una ad ogni estremità della connessione. Si crea la prima definizione di canale nel primo gestore code. Quindi, creare la seconda definizione di canale sul secondo gestore code, sull'altra estremità del link.

Entrambe le estremità devono essere definite utilizzando lo stesso nome canale. Le due estremità devono avere tipi di canale compatibili, ad esempio: mittente e destinatario.

Per creare una definizione di canale per un'estremità del collegamento utilizzare il comando MQSC `DEFINE CHANNEL`. Includere il nome del canale, il tipo di canale per questa estremità della connessione, un nome connessione, una descrizione (se richiesto), il nome della coda di trasmissione (se richiesto) e il protocollo di trasmissione. Includere anche qualsiasi altro attributo che si desidera sia diverso dai valori predefiniti di sistema per il tipo di canale richiesto, utilizzando le informazioni raccolte in precedenza.

Viene fornito un aiuto per decidere i valori degli attributi di canale in [Attributi di canale](#).




Nota: Si consiglia di denominare tutti i canali nella rete in modo univoco. L'inclusione dei nomi dei gestori code di origine e di destinazione nel nome del canale è un buon modo per farlo.

Crea esempio di canale

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +
DESCR('Sender channel to QM2') +
CONNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

In tutti gli esempi di MQSC, il comando viene visualizzato come appare in un file di comandi e come viene immesso in AIX, Linux, and Windows. I due metodi sono identici, tranne che per emettere un comando in modo interattivo, è necessario prima avviare una sessione MQSC. Immettere `runmqsc`, per il gestore code predefinito o `runmqsc qmname` dove `qmname` è il nome del gestore code richiesto. Quindi immettere qualsiasi numero di comandi, come mostrato negli esempi.

Per la portabilità, limitare la lunghezza della linea dei comandi a 72 caratteri. Utilizzare il carattere di concatenazione, `+`, come mostrato per continuare su più di una riga:

-  Su Windows utilizzare Ctrl - z per terminare la voce sulla riga comandi.
-   Su AIX and Linux, utilizzare Ctrl - d.
- In alternativa, su AIX, Linux, and Windows, utilizzare il comando **end**.

Visualizzazione di un canale

Utilizzare il comando MQSC `DISPLAY CHANNEL` per visualizzare gli attributi di un canale.

Il parametro `ALL` del comando `DISPLAY CHANNEL` viene assunto per impostazione predefinita se non sono richiesti attributi specifici e il nome del canale specificato non è generico.

Gli attributi sono descritti in [Attributi del canale](#).

Visualizza esempi di canale


```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE, CONVERT
DISPLAY CHANNEL(QM1.TO.*) TRPTYPE, CONVERT
DISPLAY CHANNEL(*) TRPTYPE, CONVERT
DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

Visualizzazione dello stato del canale

Utilizzare il comando MQSC `DISPLAY CHSTATUS`, specificando il nome del canale e se si desidera lo stato corrente dei canali o lo stato delle informazioni salvate.

`DISPLAY CHSTATUS` si applica a tutti i canali di messaggi. Non si applica a canali MQI diversi dai canali di connessione server.

Le informazioni visualizzate includono:

- Nome canale
- Nome connessione di comunicazione
- Stato in dubbio del canale (dove appropriato)
- Ultimo numero sequenza
- Nome coda di trasmissione (se appropriato)
- L'identificativo in dubbio (dove appropriato)
- L'ultimo numero di sequenza con commit
- Identificativo LUW (Logical unit of work)
- Processo ID
-  ID thread (solo per Windows)

Visualizza esempi di stato del canale

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

Lo stato salvato non si applica fino a quando non viene trasmesso almeno un batch di messaggi sul canale. Lo stato viene salvato anche quando un canale viene arrestato (utilizzando il comando STOP CHL) e quando il gestore code viene chiuso.

Controllo dei collegamenti mediante ping

Utilizzare il comando MQSC **PING CHANNEL** per scambiare un messaggio di dati fisso con l'estremità remota.

Il ping fornisce al supervisore del sistema la certezza che il collegamento sia disponibile e funzionante.

Il ping non comporta l'utilizzo di code di trasmissione e code di destinazione. Utilizza le definizioni di canale, il link di comunicazione correlato e l'impostazione di rete. Può essere utilizzato solo se il canale non è attualmente attivo.

È disponibile solo dai canali mittente, server e mittente cluster. Il canale corrispondente viene avviato sul lato opposto del link ed esegue la negoziazione del parametro di avvio. Gli errori vengono notificati normalmente.

Il risultato dello scambio di messaggi viene presentato come Ping complete o come un messaggio di errore.

Ping con LU 6.2

Quando viene richiamato il ping, per impostazione predefinita nessun ID utente o password viene inoltrato all'estremità di ricezione. Se l'ID utente e la password sono richiesti, possono essere creati all'estremità iniziale nella definizione del canale. Se una password viene immessa nella definizione di canale, viene codificata da IBM MQ prima di essere salvata. Viene quindi decodificato prima di scorrere attraverso la conversazione.

Attività correlate

Utilizzo del ping per verificare le comunicazioni

[Ping di un canale per la verifica di una connessione](#)

Riferimenti correlati

[PING CHANNEL \(verifica risposta canale\)](#)

Avvio di un canale

Utilizzare il comando MQSC START CHANNEL per i canali mittente, server e richiedente. Per consentire alle applicazioni di scambiare messaggi, è necessario avviare un programma listener per le connessioni in entrata.

START CHANNEL non è necessario quando un canale è stato configurato con l'attivazione del gestore code.

Quando avviato, l'MCA mittente legge le definizioni di canale e apre la coda di trasmissione. Viene emessa una sequenza di avvio del canale, che avvia in remoto l'MCA corrispondente del canale ricevente o server. Una volta avviati, i processi del mittente e del server attendono i messaggi in arrivo sulla coda di trasmissione e li trasmettono non appena arrivano.

Quando si utilizzano i canali di attivazione o di esecuzione come thread, assicurarsi che l'iniziatore di canali sia disponibile per monitorare la coda di avvio. L'iniziatore di canali viene avviato per impostazione predefinita come parte del gestore code.

Tuttavia, TCP e LU 6.2 forniscono altre funzioni:

- ▶ **Linux** ▶ **AIX** Per TCP su AIX and Linux, inetd può essere configurato per avviare un canale. inetd viene avviata come processo separato.
- ▶ **Linux** ▶ **AIX** Per LU 6.2 in AIX and Linux, configurare il proprio prodotto SNA per avviare il processo responder LU 6.2 .
- ▶ **Windows** Per LU 6.2 in Windows, utilizzando SNA Server è possibile utilizzare TpStart (un programma di utilità fornito con SNA Server) per avviare un canale. TpStart viene avviato come processo separato.

L'utilizzo dell'opzione Start provoca sempre la risincronizzazione del canale, se necessario.

Perché l'inizio abbia successo:

- Le definizioni di canale, locale e remoto, devono esistere. Se non esiste una definizione di canale appropriata per un canale ricevente o di connessione server, ne viene creata automaticamente una predefinita se il canale è definito automaticamente. Vedere [Programma di uscita di definizione automatica del canale](#).
- La coda di trasmissione deve esistere e non deve essere utilizzata da altri canali.
- Gli MCA, locali e remoti, devono esistere.
- Il collegamento di comunicazione deve essere disponibile.
- I gestori code devono essere in esecuzione, locali e remoti.
- Il canale dei messaggi non deve essere già in esecuzione.

Viene restituito un messaggio sullo schermo che conferma che la richiesta di avviare un canale è stata accettata. Per confermare che il comando di avvio ha avuto esito positivo, controllare il log degli errori oppure utilizzare DISPLAY CHSTATUS. I log degli errori sono:

▶ **Windows** **Windows**

MQ_DATA_PATH\qmgrs\qmname\errors\AMQERR01. LOG (per ogni gestore code denominato qmname)

MQ_DATA_PATH\qmgrs\@SYSTEM\errors\AMQERR01. LOG (per errori generali)

MQ_DATA_PATH rappresenta la directory di alto livello in cui è installato IBM MQ .

Nota: Su Windows, si riceve ancora un messaggio nel registro eventi dell'applicazione dei sistemi Windows .

▶ **Linux** ▶ **AIX** **AIX and Linux**

/var/mqm/qmgrs/qmname/errors/AMQERR01. LOG (per ogni gestore code denominato qmname)

/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01. LOG (per errori generali)

Su AIX, Linux, and Windows, utilizzare il comando **runmqclsr** per avviare il processo listener IBM MQ . Per impostazione predefinita, le richieste in entrata per l'allegato del canale fanno sì che il processo listener avvii gli MCA come thread del processo amqrmpa.

```
runmqclsr -t tcp -m QM2
```

Per le connessioni in entrata, è necessario avviare il canale in uno dei tre seguenti modi:

1. Utilizzare il comando MQSC START CHANNEL, specificando il nome del canale, per avviare il canale come un processo o un thread, in base al parametro MCATYPE. (Se i canali vengono avviati come thread, sono thread di un iniziatore di canali.)

```
START CHANNEL(QM1.TO.QM2)
```

2. Utilizzare il comando di controllo runmqchl per avviare un canale come processo.

```
runmqchl -c QM1.TO.QM2 -m QM1
```

3. Utilizzare l'iniziatore del canale per attivare il canale.

ALW Arresto di un canale

Utilizzare il comando MQSC STOP CHANNEL per richiedere al canale di arrestare l'attività. Il canale non avvia un nuovo batch di messaggi finché l'operatore non avvia nuovamente il canale.

Per informazioni sul riavvio dei canali arrestati, consultare [“Riavvio dei canali arrestati”](#) a pagina 246.

Questo comando può essere emesso su un canale di qualsiasi tipo, ad eccezione di MQCHT_CLNTCONN.

È possibile selezionare il tipo di arresto richiesto:

Esempio di arresto del quiesce

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

Questo comando richiede la chiusura ordinata del canale. Il batch di messaggi corrente viene completato e la procedura del punto di sincronizzazione viene eseguita con l'altra estremità del canale. Se il canale è inattivo, questo comando non termina un canale di ricezione.

Esempio di arresto forzato

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

Questa opzione arresta il canale immediatamente, ma non termina il thread o il processo del canale. Il canale non completa l'elaborazione del batch di messaggi corrente e può, quindi, lasciare il canale in dubbio. In generale, considerare l'utilizzo dell'opzione di arresto della sospensione.

Esempio di arresto

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

Questa opzione arresta immediatamente il canale e termina il thread o il processo del canale.

Esempio di arresto (sospensione) arrestato

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

Questo comando non specifica un MODE, quindi il valore predefinito è MODE (QUIESCE). Richiede che il canale sia arrestato in modo che non possa essere riavviato in modo automatico, ma deve essere avviato manualmente.

Esempio di arresto (sospensione) inattivo

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

Questo comando non specifica un MODE, quindi il valore predefinito è MODE (QUIESCE). Richiede che il canale venga reso inattivo in modo che venga riavviato automaticamente quando richiesto.

ALW Ridenominazione di un canale

Utilizzare MQSC per ridenominare un canale di messaggi.

Utilizzare MQSC per effettuare le operazioni riportate di seguito:

1. Utilizzare STOP CHANNEL per arrestare il canale.
2. Utilizzare DEFINE CHANNEL per creare una definizione di canale duplicata con il nuovo nome.
3. Utilizzare DISPLAY CHANNEL per verificare che sia stato creato correttamente.
4. Utilizzare DELETE CHANNEL per eliminare la definizione del canale originale.

Se si decide di ridenominare un canale di messaggi, ricordare che un canale ha due definizioni di canale, una a ciascuna estremità. Assicurarsi di rinominare il canale ad entrambe le estremità contemporaneamente.

Reimpostazione di un canale

Utilizzare il comando MQSC RESET CHANNEL per modificare il numero di sequenza del messaggio.

Il comando RESET CHANNEL è disponibile per qualsiasi canale di messaggi, ma non per i canali MQI (connessione client o connessione server). Il primo messaggio avvia la nuova sequenza al successivo avvio del canale.

Se il comando viene emesso su un canale mittente o server, informa l'altro lato della modifica quando il canale viene riavviato.

Concetti correlati

[“Introduzione agli oggetti” a pagina 261](#)

I canali devono essere definiti e i relativi oggetti associati devono esistere ed essere disponibili per l'utilizzo, prima che un canale possa essere avviato. Questa sezione mostra come.

[“Funzione di controllo canale” a pagina 235](#)

La funzione di controllo del canale consente di definire, monitorare e controllare i canali.

Attività correlate

[“Configurazione dell'accodamento distribuito” a pagina 205](#)

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ, incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

Riferimenti correlati

[Reimpostazione canale](#)

Risoluzione dei messaggi in dubbio su un canale

Utilizzare il comando MQSC RESOLVE CHANNEL quando i messaggi sono in dubbio da parte di un mittente o di un server. Ad esempio, perché un'estremità del link è stata terminata e non vi è alcuna prospettiva di ripristino.

Il comando [RESOLVE CHANNEL](#) accetta uno dei due parametri: BACKOUT o COMMIT. Il backout ripristina i messaggi nella coda di trasmissione, mentre il commit li elimina.

Il programma del canale non tenta di stabilire una sessione con un partner. Invece, determina l'identificativo LUWID (logical unit of work identifier) che rappresenta i messaggi in dubbio. Quindi, come richiesto, emette:

- BACKOUT per ripristinare i messaggi nella coda di trasmissione; oppure
- COMMIT per cancellare i messaggi dalla coda di trasmissione.

Affinché la risoluzione abbia successo:

- Il canale deve essere inattivo
- Il canale deve essere in dubbio
- Il tipo di canale deve essere mittente, server o mittente cluster
- È necessario che esista una definizione di canale locale
- Il gestore code locale deve essere in esecuzione

Concetti correlati

[“Introduzione agli oggetti” a pagina 261](#)

I canali devono essere definiti e i relativi oggetti associati devono esistere ed essere disponibili per l'utilizzo, prima che un canale possa essere avviato. Questa sezione mostra come.

[“Funzione di controllo canale” a pagina 235](#)

La funzione di controllo del canale consente di definire, monitorare e controllare i canali.

Attività correlate

[“Configurazione dell'accodamento distribuito” a pagina 205](#)

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

Riferimenti correlati



[Risoluzione canale](#)

Impostazione della comunicazione su Windows

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché ciò abbia esito positivo, la connessione deve essere definita e disponibile. Questa sezione spiega come eseguire questa operazione utilizzando i moduli di comunicazione disponibili per sistemi IBM MQ for Windows .

Prima di iniziare

Potrebbe essere utile fare riferimento a [Configurazione di esempio - IBM MQ for Windows](#).

  Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linux o Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Informazioni su questa attività

Quando si configura la comunicazione per IBM MQ su Windows, è possibile scegliere tra i seguenti tipi di comunicazione:

- TCP/IP
- LU 6.2
- NetBIOS

Procedura

- Per informazioni sull'impostazione della comunicazione per il sistema Windows , consultare l'argomento secondario per il tipo di comunicazione scelto:
 - [“Definizione di un collegamento TCP su Windows” a pagina 269](#)
 - [“Definizione di una connessione LU 6.2 su Windows” a pagina 271](#)
 - [“Definizione di una connessione NetBIOS su Windows” a pagina 272](#)

Non tutte le funzioni e le funzioni di IBM MQ for Windows sono disponibili in ambienti che utilizzano protocolli di comunicazioni diversi da TCP/IP. L'elemento non disponibile è IBM MQ Explorer.

Attività correlate

[“Monitoraggio e controllo dei canali su AIX, Linux, and Windows” a pagina 258](#)

Per DQM è necessario creare, monitorare e controllare i canali per i gestori code remoti. È possibile controllare i canali utilizzando comandi, programmi, IBM MQ Explorer, file per le definizioni dei canali e un'area di memoria per le informazioni di sincronizzazione.

[“Configurazione delle connessioni tra client e server” a pagina 16](#)

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

[“Impostazione della comunicazione su AIX and Linux” a pagina 275](#)

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché ciò abbia esito positivo, la connessione deve essere definita e disponibile. Questa sezione spiega come eseguire questa operazione utilizzando i moduli di comunicazione disponibili per sistemi IBM MQ for UNIX or Linux .

Riferimenti correlati

[“Il tipo di comunicazione da utilizzare” a pagina 17](#)

Piattaforme differenti supportano protocolli di comunicazione differenti. La scelta del protocollo di trasmissione dipende dalla combinazione di IBM MQ MQI client e delle piattaforme server.

Windows *Definizione di un collegamento TCP su Windows*

Definire una connessione TCP configurando un canale all'estremità di invio per specificare l'indirizzo della destinazione ed eseguendo un programma listener all'estremità di ricezione.

Prima di iniziare

MQ Adv. **CD** Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linuxo Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Fine invio

Specificare il nome host o l'indirizzo TCP della macchina di destinazione nel campo Nome connessione della definizione di canale.

La porta su cui connettersi è il valore predefinito 1414. Il numero di porta 1414 viene assegnato da Internet Assigned Numbers Authority a IBM MQ.

Per utilizzare un numero di porta diverso da quello predefinito, specificarlo nel campo del nome della connessione della definizione dell'oggetto canale:

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +
    TRPTYPE(TCP) +
    CONNAME('OS2ROG3(1822)') +
    XMITQ('XMitQ name') +
    REPLACE
```

dove OS2ROG3 è il nome DNS del gestore code remoto e 1822 è la porta richiesta. (deve essere la porta su cui è in ascolto il listener all'estremità di ricezione).

Un canale in esecuzione deve essere arrestato e riavviato per acquisire qualsiasi modifica alla definizione dell'oggetto del canale.

È possibile modificare il numero di porta predefinito specificandolo nel file `.ini` per IBM MQ for Windows:

```
TCP:
Port=1822
```

Nota: Per selezionare quale numero di porta TCP/IP utilizzare, IBM MQ utilizza il primo numero di porta che trova nella seguente sequenza:

1. Il numero di porta specificato esplicitamente nella definizione del canale o nella riga comandi. Questo numero consente la sovrascrittura del numero di porta predefinito per un canale.
2. L'attributo port specificato nella stanza TCP del file `.ini`. Questo numero consente al numero di porta predefinito di essere sovrascritto per un gestore code.
3. Il valore predefinito è 1414. Questo è il numero assegnato a IBM MQ da Internet Assigned Numbers Authority per le connessioni in entrata e in uscita.

Per ulteriori informazioni sui valori impostati utilizzando `qm.ini`, consultare [Stanza del file di configurazione per l'accodamento distribuito](#).

Ricezione su TCP

Per avviare un programma del canale ricevente, è necessario avviare un programma listener per rilevare le richieste di rete in entrata e avviare il canale associato. È possibile utilizzare il listener IBM MQ.

I programmi del canale ricevente vengono avviati in risposta a una richiesta di avvio dal canale mittente.

Per avviare un programma del canale ricevente, è necessario avviare un programma listener per rilevare le richieste di rete in entrata e avviare il canale associato. È possibile utilizzare il listener IBM MQ.

Per eseguire il listener fornito con IBM MQ, che avvia nuovi canali come thread, utilizzare il comando `runmqlsr`.

Un esempio di base dell'utilizzo del comando `runmqlsr`:

```
runmqlsr -t tcp [-m QMNAME] [-p 1822]
```

Le parentesi quadre indicano parametri facoltativi; `QMNAME` non è richiesto per il gestore code predefinito e il numero di porta non è richiesto se si utilizza il valore predefinito (1414). Il numero di porta non deve essere superiore a 65535.

Nota: Per selezionare quale numero di porta TCP/IP utilizzare, IBM MQ utilizza il primo numero di porta che trova nella seguente sequenza:

1. Il numero di porta specificato esplicitamente nella definizione del canale o nella riga comandi. Questo numero consente la sovrascrittura del numero di porta predefinito per un canale.
2. L'attributo port specificato nella stanza TCP del file `.ini`. Questo numero consente al numero di porta predefinito di essere sovrascritto per un gestore code.
3. Il valore predefinito è 1414. Questo è il numero assegnato a IBM MQ da Internet Assigned Numbers Authority per le connessioni in entrata e in uscita.

Per prestazioni ottimali, eseguire il listener IBM MQ come un'applicazione attendibile come descritto in [“Esecuzione di canali e listener come applicazioni attendibili” a pagina 257](#). Per informazioni sulle applicazioni attendibili, consultare [Limitazioni per le applicazioni attendibili](#)

Utilizzo dell'opzione TCP/IP SO_KEEPALIVE

Se si desidera utilizzare l'opzione Windows `SO_KEEPALIVE`, è necessario aggiungere la seguente voce al registro:

```
TCP:  
KeepAlive=yes
```

Per ulteriori informazioni sull'opzione `SO_KEEPALIVE`, consultare [“Verifica che l'altra estremità del canale sia ancora disponibile” a pagina 242](#).

Su Windows, il valore di registro

`HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` per l'opzione Windows

KeepAliveTime controlla l'intervallo che trascorre prima che venga controllata la connessione. Il valore predefinito è due ore.

Utilizzo dell'opzione backlog del listener TCP

In TCP, le connessioni sono considerate incomplete a meno che non si verifichi un handshake a tre vie tra il server e il client. Queste connessioni vengono chiamate richieste di connessione in sospeso. Viene impostato un valore massimo per queste richieste di connessione in sospeso e può essere considerato un backlog di richieste in attesa sulla porta TCP affinché il listener accetti la richiesta.

Consultare “Utilizzo dell'opzione di backlog del listener TCP su IBM MQ for Multiplatforms” a pagina 279 per ulteriori informazioni e il valore specifico per Windows.

Windows Definizione di una connessione LU 6.2 su Windows

SNA deve essere configurato in modo che sia possibile stabilire una conversazione LU 6.2 tra le macchine.

Una volta configurato SNA, procedere nel modo seguente.

Consultare la seguente tabella per informazioni.

Piattaforma remota	TPNAME	TPPATH
z/OS o MVS/ESA senza CICS	Le stesse informazioni del lato corrispondente sul gestore code remoto.	-
z/OS o MVS/ESA utilizzando CICS	CKRC (mittente) CKSV (richiedente) CKRC (server)	-
IBM i	Uguale al valore di confronto nella specifica di instradamento sul sistema IBM i .	-
Sistemi AIX and Linux	Le stesse informazioni del lato corrispondente sul gestore code remoto.	<i>MQ_INSTALLATION_PATH</i> /bin/amqcrs6a
Windows	Come specificato nel comando Windows Esegui listener o nel programma di transazione richiamabile definito utilizzando TpSetup su Windows.	<i>MQ_INSTALLATION_PATH</i> \bin\amqcrs6a

MQ_INSTALLATION_PATH rappresenta la directory di livello superiore in cui è installato IBM MQ .

Se si dispone di più di un gestore code sulla stessa macchina, verificare che i nomi TP nelle definizioni di canale siano univoci.

Per informazioni più recenti sulla configurazione di AnyNet SNA su TCP/IP, consultare la seguente documentazione in linea IBM : [AnyNet SNA su TCP/IP](#) e [SNA Node Operations](#).

Concetti correlati

“Invio fine su LU 6.2 su Windows” a pagina 271

Creare un oggetto lato CPI-C (destinazione simbolica) dall'applicazione di gestione del prodotto LU 6.2 che si sta utilizzando. Immettere questo nome nel campo Nome connessione nella definizione di canale. Creare anche un collegamento LU 6.2 al partner.

“Ricezione su LU 6.2 su Windows” a pagina 272

I programmi del canale ricevente vengono avviati in risposta a una richiesta di avvio dal canale mittente.

Windows Invio fine su LU 6.2 su Windows

Creare un oggetto lato CPI-C (destinazione simbolica) dall'applicazione di gestione del prodotto LU 6.2 che si sta utilizzando. Immettere questo nome nel campo Nome connessione nella definizione di canale. Creare anche un collegamento LU 6.2 al partner.

Nell'oggetto lato CPI-C, immettere il nome LU partner sulla macchina ricevente, il nome TP e il nome modo. Ad esempio:

Partner LU Name	OS2R0G2
Partner TP Name	recv
Mode Name	#INTER

Windows *Ricezione su LU 6.2 su Windows*

I programmi del canale ricevente vengono avviati in risposta a una richiesta di avvio dal canale mittente.

Per avviare un programma del canale di ricezione, è necessario avviare un programma listener per rilevare le richieste di rete in entrata e avviare il canale associato. Si avvia questo programma listener con il comando RUNMQLSR, fornendo il TpName su cui eseguire l'ascolto. In alternativa, è possibile utilizzare TpStart in SNA Server per Windows.

Utilizzo del comando RUNMQLSR

Esempio del comando per avviare il listener:

```
RUNMQLSR -t LU62 -n RECV -m QMNAME
```

dove RECV è il TpName specificato all'altra estremità (di invio) come "TpName da avviare sul lato remoto". Il parametro **-m** utilizzato nell'ultima parte di questo comando è facoltativo e non è richiesto per il gestore code predefinito.

È possibile avere più di un gestore code in esecuzione su una macchina. È necessario assegnare un TpName differente a ogni gestore code, quindi avviare un programma listener per ogni gestore code. Ad esempio:

```
RUNMQLSR -t LU62 -m QM1 -n TpName1  
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

Per prestazioni ottimali, eseguire il listener IBM MQ come un'applicazione attendibile, come descritto in [Esecuzione di canali e listener come applicazioni attendibili](#). Consultare [Limitazioni per le applicazioni attendibili](#) per informazioni sulle applicazioni attendibili.

È possibile arrestare tutti i listener IBM MQ in esecuzione su un gestore code inattivo, utilizzando il comando:

```
ENDMQLSR -m QMNAME
```

Utilizzo di Microsoft SNA Server su Windows

È possibile utilizzare TpSetup (da SNA Server SDK) per definire un TP richiamabile che poi guida amqcrs6a.exe, oppure è possibile impostare manualmente diversi valori di registro. I parametri che devono essere passati a amqcrs6a.exe sono:

```
-m QM -n TpName
```

dove *QM* è il nome del gestore code e *TpName* è il nome TP. Per ulteriori informazioni, consultare il manuale *Microsoft SNA Server APPC Programmers Guide* o il manuale *Microsoft SNA Server CPI-C Programmers Guide*.

Se non si specifica un nome gestore code, viene utilizzato il gestore code predefinito.

Windows *Definizione di una connessione NetBIOS su Windows*

Una connessione NetBIOS si applica solo a un client e a un server su cui è in esecuzione Windows. IBM MQ utilizza tre tipi di risorsa NetBIOS quando stabilisce una connessione NetBIOS a un altro prodotto IBM MQ : sessioni, comandi e nomi. Ciascuna di queste risorse ha un limite, che viene stabilito per impostazione predefinita o per scelta durante l'installazione di NetBIOS.

Ogni canale in esecuzione, indipendentemente dal tipo, utilizza una sessione NetBIOS e un comando NetBIOS . L'implementazione IBM NetBIOS consente a più processi di utilizzare lo stesso nome NetBIOS locale. Pertanto, solo un nome NetBIOS deve essere disponibile per l'utilizzo da parte di IBM MQ. Le implementazioni di altri fornitori, ad esempio l'emulazione NetBIOS di Novell, richiedono un nome locale differente per processo. Verificare i requisiti dalla documentazione per il prodotto NetBIOS che si sta utilizzando.

In tutti i casi, assicurarsi che siano già disponibili risorse sufficienti di ciascun tipo o aumentare i valori massimi specificati nella configurazione. Qualsiasi modifica ai valori richiede un riavvio del sistema.

Durante l'avvio del sistema, il driver di periferica NetBIOS visualizza il numero di sessioni, comandi e nomi disponibili per l'utilizzo da parte delle applicazioni. Queste risorse sono disponibili per tutte le applicazioni basate sul NetBIOS in esecuzione sullo stesso sistema. Pertanto, è possibile che altre applicazioni utilizzino queste risorse prima che IBM MQ le acquisisca. L'amministratore della rete LAN dovrebbe essere in grado di chiarire questo.

Concetti correlati

[“Definizione del nome IBM MQ locale NetBIOS” a pagina 273](#)

Il nome NetBIOS locale utilizzato dai processi del canale IBM MQ può essere specificato in tre modi.

[“Definizione dei limiti di nome, comando e sessione NetBIOS del gestore code” a pagina 274](#)

I limiti del gestore code per sessioni, comandi e nomi NetBIOS possono essere specificati in due modi.

[“Stabilire il numero dell'adattatore LAN” a pagina 274](#)

Affinché i canali funzionino correttamente su NetBIOS, è necessario che il supporto dell'adattatore a ciascuna estremità sia compatibile. IBM MQ consente di controllare il numero dell'adattatore LAN (LANA) utilizzando il valore AdapterNum nella stanza NETBIOS del file qm.ini e specificando il parametro **-a** nel comando `runmqtsr`.

[“Avvio della connessione NetBIOS” a pagina 274](#)

Definizione delle fasi necessarie per avviare una connessione.

[“Definizione del listener di destinazione per la connessione NetBIOS” a pagina 275](#)

Definizione delle operazioni da eseguire all'estremità ricevente della connessione NetBIOS .

Windows Definizione del nome IBM MQ locale NetBIOS

Il nome NetBIOS locale utilizzato dai processi del canale IBM MQ può essere specificato in tre modi.

In ordine di precedenza, i tre modi sono:

1. Il valore specificato nel parametro **-l** del comando `runmqtsr` , ad esempio:

```
runmqtsr -t netbios -l my_station
```

2. La variabile di ambiente **MQNAME** con un valore stabilito dal comando:

```
SET MQNAME= my_station
```

Ad esempio:

```
SET MQNAME=CLIENT1
```

È possibile impostare il valore **MQNAME** per ogni processo. In alternativa, è possibile impostarlo a livello di sistema nel registro di Windows .

Se si utilizza un'implementazione NetBIOS che richiede nomi univoci, è necessario immettere un comando **SET MQNAME** in ogni finestra in cui viene avviato un processo IBM MQ . Il valore **MQNAME** è arbitrario ma deve essere univoco per ogni processo.

3. La [stanza NETBIOS](#) nel file di configurazione del gestore code qm . ini. Ad esempio:

```
NETBIOS:
```

```
LocalName= my_station
```

Nota:

1. a causa delle variazioni nell'implementazione dei prodotti NetBIOS supportati, si consiglia di rendere ciascun nome NetBIOS univoco nella rete. In caso contrario, potrebbero verificarsi risultati imprevedibili. Se si verificano dei problemi nello stabilire un canale NetBIOS e nel log degli errori del gestore code sono presenti dei messaggi di errore che mostrano un codice di ritorno NetBIOS X'15 ', rivedere l'utilizzo dei nomi NetBIOS .
2. Su Windows, non è possibile utilizzare il nome della macchina come nome NetBIOS perché Windows lo utilizza già.
3. L'avvio del canale mittente richiede che venga specificato un nome NetBIOS utilizzando la variabile di ambiente MQNAME o il LocalName nel file qm.ini .

Windows *Definizione dei limiti di nome, comando e sessione NetBIOS del gestore code*

I limiti del gestore code per sessioni, comandi e nomi NetBIOS possono essere specificati in due modi.

In ordine di precedenza, questi modi sono:

1. I valori specificati nel comando RUNMQLSR:

```
-s Sessions  
-e Names  
-o Commands
```

Se l'operando -m non viene specificato nel comando, i valori si applicano solo al gestore code predefinito.

2. La stanza NETBIOS nel file di configurazione del gestore code qm.ini. Ad esempio:

```
NETBIOS:  
NumSess= Qmgr_max_sess  
NumCmds= Qmgr_max_cmds  
NumNames= Qmgr_max_names
```

Windows *Stabilire il numero dell'adattatore LAN*

Affinché i canali funzionino correttamente su NetBIOS, è necessario che il supporto dell'adattatore a ciascuna estremità sia compatibile. IBM MQ consente di controllare il numero dell'adattatore LAN (LANA) utilizzando il valore AdapterNum nella stanza NETBIOS del file qm.ini e specificando il parametro **-a** nel comando runmqsr.

Il numero di adattatore LAN predefinito utilizzato da IBM MQ per connessioni NetBIOS è 0. Verificare il numero utilizzato sul sistema nel modo seguente:

Su Windows, non è possibile interrogare il numero dell'adattatore LAN direttamente tramite il sistema operativo. Utilizzare invece LANACFG.EXE , disponibile da Microsoft. L'output dello strumento visualizza i numeri dell'adattatore LAN virtuale e i relativi collegamenti effettivi. Per ulteriori informazioni sui numeri degli adattatori LAN, consultare l'articolo della Microsoft Knowledge Base 138037 *HOWTO: Use LANA Numbers in a 32 - bit Environment*.

Specificare il valore corretto per la stanza NETBIOS del file di configurazione del gestore code, qm.ini:

```
NETBIOS:  
AdapterNum= n
```

dove n è il numero dell'adattatore LAN corretto per questo sistema.

Windows *Avvio della connessione NetBIOS*

Definizione delle fasi necessarie per avviare una connessione.

Per avviare la connessione, attenersi alla seguente procedura all'estremità di invio:

1. Definire il nome della stazione NetBIOS utilizzando il valore MQNAME o LocalName .
2. Verificare il numero dell'adattatore LAN utilizzato sul sistema e specificare il file corretto utilizzando AdapterNum.
3. Nel campo ConnectionName della definizione di canale, specificare il nome NetBIOS utilizzato dal programma listener di destinazione. Su Windows, i canali NetBIOS devono essere eseguiti come thread. Eseguire questa operazione specificando MCATYPE (THREAD) nella definizione del canale.

```
DEFINE CHANNEL (cname) CHLTYPE(SDR) +  
TRPTYPE(NETBIOS) +  
CONNAME(your_station) +  
XMITQ(xmitq) +  
MCATYPE(THREAD) +  
REPLACE
```

Windows Definizione del listener di destinazione per la connessione NetBIOS

Definizione delle operazioni da eseguire all'estremità ricevente della connessione NetBIOS .

All'estremità di ricezione, seguire queste istruzioni:

1. Definire il nome della stazione NetBIOS utilizzando il valore MQNAME o LocalName .
2. Verificare il numero dell'adattatore LAN utilizzato sul sistema e specificare il file corretto utilizzando AdapterNum.
3. Definire il canale ricevente:

```
DEFINE CHANNEL (cname) CHLTYPE(RCVR) +  
TRPTYPE(NETBIOS) +  
REPLACE
```

4. Avviare il programma listener IBM MQ per stabilire la stazione e rendere possibile contattarla. Ad esempio:

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

Questo comando stabilisce your_station come una stazione NetBIOS in attesa di essere contattata. Il nome della stazione NetBIOS deve essere univoco nella rete NetBIOS .

Per prestazioni ottimali, eseguire il listener IBM MQ come un'applicazione attendibile come descritto in [“Esecuzione di canali e listener come applicazioni attendibili”](#) a pagina 257. Consultare [Limitazioni per le applicazioni attendibili](#) per informazioni sulle applicazioni attendibili.

È possibile arrestare tutti i listener IBM MQ in esecuzione su un gestore code inattivo, utilizzando il comando:

```
ENDMQLSR [-m QMNAME]
```

Se non si specifica un nome gestore code, viene utilizzato il gestore code predefinito.

Linux



AIX



Impostazione della comunicazione su AIX and Linux

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché ciò abbia esito positivo, la connessione deve essere definita e disponibile. Questa sezione spiega come eseguire questa operazione utilizzando i moduli di comunicazione disponibili per sistemi IBM MQ for UNIX or Linux .

Prima di iniziare

Potrebbe essere utile fare riferimento alle sezioni seguenti:

-  [Configurazione di esempio IBM MQ for AIX](#)
-  [Configurazione di esempio - IBM MQ per Linux](#)

  Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linuxo Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Informazioni su questa attività

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Per avere esito positivo, è necessario che la connessione sia definita e disponibile. Questa sezione spiega come eseguire questa operazione.

Quando si configura la comunicazione per IBM MQ su AIX and Linux, è possibile scegliere tra i seguenti tipi di comunicazione:



- TCP/IP
- LU 6.2

Ciascuna definizione di canale deve specificarne una solo come attributo del Protocollo di trasmissione (Tipo di trasporto). Uno o più protocolli possono essere utilizzati da un gestore code.

Per IBM MQ MQI clients, potrebbe essere utile disporre di canali alternativi che utilizzano protocolli di trasmissione differenti. Vedere [IBM MQ MQI clients](#).

Procedura

Per informazioni sull'impostazione della comunicazione per il sistema AIX o Linux , consultare l'argomento secondario per il tipo di comunicazione scelto:

- [“Definizione di un collegamento TCP su AIX and Linux” a pagina 277](#)
- [“Definizione di una connessione LU 6.2 su AIX and Linux” a pagina 280](#)
-   [“Definizione di una connessione Aspera gateway su piattaforme Linux o Windows” a pagina 874](#)

Attività correlate

[“Monitoraggio e controllo dei canali su AIX, Linux, and Windows” a pagina 258](#)

Per DQM è necessario creare, monitorare e controllare i canali per i gestori code remoti. È possibile controllare i canali utilizzando comandi, programmi, IBM MQ Explorer, file per le definizioni dei canali e un'area di memoria per le informazioni di sincronizzazione.

[“Configurazione delle connessioni tra client e server” a pagina 16](#)

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

[“Impostazione della comunicazione su Windows” a pagina 268](#)

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché ciò abbia esito positivo, la connessione deve essere definita e disponibile. Questa sezione spiega come eseguire questa operazione utilizzando i moduli di comunicazione disponibili per sistemi IBM MQ for Windows .

Riferimenti correlati

[“Il tipo di comunicazione da utilizzare” a pagina 17](#)

Piattaforme differenti supportano protocolli di comunicazione differenti. La scelta del protocollo di trasmissione dipende dalla combinazione di IBM MQ MQI client e delle piattaforme server.

La definizione del canale all'estremità di invio specifica l'indirizzo della destinazione. Il listener o il daemon inet è configurato per la connessione all'estremità di ricezione.

Prima di iniziare

Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linux o Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Fine invio

Specificare il nome host o l'indirizzo TCP della macchina di destinazione nel campo Nome connessione della definizione di canale. La porta su cui connettersi è il valore predefinito 1414. Il numero di porta 1414 viene assegnato da Internet Assigned Numbers Authority a IBM MQ.

Per utilizzare un numero di porta diverso da quello predefinito, modificare il campo del nome della connessione come segue:

```
Connection Name REMHOST(1822)
```

dove REMHOST è il nome host della macchina remota e 1822 è il numero di porta richiesto. (deve essere la porta su cui è in ascolto il listener all'estremità di ricezione).

In alternativa, è possibile modificare il numero di porta specificandolo nel file di configurazione del gestore code (qm.ini):

```
TCP:
Port=1822
```

Per ulteriori informazioni sui valori impostati utilizzando qm.ini, consultare [Stanza del file di configurazione per l'accodamento distribuito](#).

Ricezione su TCP

È possibile utilizzare il listener TCP/IP, che è il daemon inet (inetd) o il listener IBM MQ .

Alcune distribuzioni Linux utilizzano il daemon inet esteso (xinetd) invece del daemon inet. Per ulteriori informazioni su come utilizzare il daemon inet esteso su un sistema Linux , consultare il [Passo 2 di Esempio: impostazione della comunicazione tra piattaforme IBM MQ su Linux](#).

Concetti correlati

[“Utilizzo del listener TCP/IP su AIX and Linux” a pagina 278](#)

Per avviare i canali su AIX and Linux, è necessario modificare il file `/etc/services` e il file `inetd.conf`

[“Utilizzo dell'opzione di backlog del listener TCP su IBM MQ for Multiplatforms” a pagina 279](#)

In TCP, le connessioni sono considerate incomplete a meno che non si verifichi un handshake a tre vie tra il server e il client. Queste connessioni vengono chiamate richieste di connessione in sospeso. Viene impostato un valore massimo per queste richieste di connessione in sospeso e può essere considerato un backlog di richieste in attesa sulla porta TCP affinché il listener accetti la richiesta.

[“Utilizzo del listener IBM MQ” a pagina 279](#)

Per eseguire il listener fornito con IBM MQ, che avvia nuovi canali come thread, utilizzare il comando `runmq1sr`.

[“Utilizzo dell'opzione TCP/IP SO_KEEPALIVE” a pagina 280](#)

Su alcuni sistemi AIX and Linux , è possibile definire il tempo di attesa TCP prima di verificare che la connessione sia ancora disponibile e la frequenza con cui tenta nuovamente la connessione se il primo

controllo ha esito negativo. Questo è un parametro ottimizzabile del kernel o può essere immesso sulla riga comandi.

Linux > AIX Utilizzo del listener TCP/IP su AIX and Linux

Per avviare i canali su AIX and Linux, è necessario modificare il file `/etc/services` e il file `inetd.conf`

Seguire queste istruzioni:

1. Modificare il file `/etc/services`.

Nota: Per modificare il file `/etc/services`, è necessario essere collegati come superutente o root. È possibile modificarlo, ma deve corrispondere al numero di porta specificato all'estremità di invio.

Aggiungere la seguente riga al file:

```
MQSeries 1414/tcp
```

dove 1414 è il numero di porta richiesto da IBM MQ. Il numero di porta non deve essere superiore a 65535.

2. Aggiungere una riga nel file `inetd.conf` per richiamare il programma `amqcrsta`, dove `MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ :

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m Queue_Man_Name]
```

Gli aggiornamenti sono attivi dopo che `inetd` ha riletto i file di configurazione. Per eseguire questa operazione, immettere i seguenti comandi dall'ID utente root:

- **AIX** Su AIX:

```
refresh -s inetd
```

- **Linux** Su sistemi Linux:

```
kill -1 process_number
```

Quando il programma listener avviato da `inetd` eredita la locale da `inetd`, è possibile che MQMDE non venga rispettato (unito) e che venga inserito nella coda come dati del messaggio. Per assicurarsi che MQMDE sia rispettato, è necessario impostare correttamente la locale. La locale impostata da `inetd` potrebbe non corrispondere a quella scelta per altre locale utilizzate dai processi IBM MQ . Per impostare la locale:

1. Creare uno script di shell che imposta le variabili di ambiente `LANG`, `LC_COLLATE`, `LC_CTYPE`, `LC_MONETARY`, `LC_NUMERIC`, `LC_TIME` e `LC_MESSAGES` sulla locale utilizzata per altri processi IBM MQ .
2. Nello stesso script di shell, richiamare il programma listener.
3. Modificare il file `inetd.conf` per richiamare lo script shell al posto del programma listener.

È possibile avere più di un gestore code sul server. È necessario aggiungere una riga a ciascuno dei due file, per ciascuno dei gestori code. Ad esempio:

```
MQSeries1 1414/tcp  
MQSeries2 1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

Dove `MQ_INSTALLATION_PATH` rappresenta la directory di alto livello in cui è installato IBM MQ .

Ciò evita la generazione di messaggi di errore se esiste una limitazione sul numero di richieste di connessione in sospeso accodate su una singola porta TCP. Per informazioni sul numero di richieste di connessione in sospeso, vedere [“Utilizzo dell'opzione di backlog del listener TCP su IBM MQ for Multiplatforms”](#) a pagina 279.

Multi Utilizzo dell'opzione di backlog del listener TCP su IBM MQ for Multiplatforms

In TCP, le connessioni sono considerate incomplete a meno che non si verifichi un handshake a tre vie tra il server e il client. Queste connessioni vengono chiamate richieste di connessione in sospeso. Viene impostato un valore massimo per queste richieste di connessione in sospeso e può essere considerato un backlog di richieste in attesa sulla porta TCP affinché il listener accetti la richiesta.

I valori di backlog del listener predefiniti vengono mostrati in [Tabella 24 a pagina 279](#).

<i>Tabella 24. Numero massimo di richieste di connessione in sospeso accodate su una porta TCP/IP</i>	
Piattaforma server	Numero massimo di richieste di connessione
AIX AIX	100
Linux Linux	100
IBM i IBM i	255
Windows ServerWindows	100

Se il backlog raggiunge i valori mostrati in [Tabella 24 a pagina 279](#), la connessione TCP/IP viene rifiutata e non è possibile avviare il canale.

Per i canali MCA, ciò fa sì che il canale entri in uno stato RETRY e riprovi la connessione in un secondo momento.

Tuttavia, per evitare questo errore, è possibile aggiungere una voce nel file `qm.ini` :

```
TCP:
ListenerBacklog = n
```

Ciò sovrascrive il numero massimo predefinito di richieste in sospeso (consultare [Tabella 24 a pagina 279](#)) per il listener TCP/IP.

Nota: Alcuni sistemi operativi supportano un valore maggiore di quello predefinito. Se necessario, questo valore può essere utilizzato per evitare di raggiungere il limite di connessione.

Per eseguire il listener con l'opzione `backlog` abilitata:

- Utilizzare il comando `runmqtsr -b` oppure
- Utilizzare il comando MQSC **DEFINE LISTENER** con l'attributo `BACKLOG` impostato sul valore richiesto.

Per informazioni sul comando `runmqtsr`, consultare [runmqtsr](#). Per informazioni sul comando `DEFINE LISTENER`, consultare [DEFINE LISTENER](#).

Concetti correlati

[“Using the TCP listener backlog option on z/OS”](#) a pagina 984

When receiving on TCP/IP, a maximum number of outstanding connection requests is set. These outstanding requests can be considered a *backlog* of requests waiting on the TCP/IP port for the listener to accept the request.

Linux **AIX** Utilizzo del listener IBM MQ

Per eseguire il listener fornito con IBM MQ, che avvia nuovi canali come thread, utilizzare il comando `runmqtsr`.

Ad esempio:

```
runmqclsr -t tcp [-m QMNAME] [-p 1822]
```

Le parentesi quadre indicano parametri facoltativi; QMNAME non è richiesto per il gestore code predefinito e il numero di porta non è richiesto se si utilizza il valore predefinito (1414). Il numero di porta non deve essere superiore a 65535.

Per prestazioni ottimali, eseguire il listener IBM MQ come un'applicazione attendibile come descritto in [“Esecuzione di canali e listener come applicazioni attendibili”](#) a pagina 257. Consultare [Limitazioni per le applicazioni attendibili](#) per informazioni sulle applicazioni attendibili.

È possibile arrestare tutti i listener IBM MQ in esecuzione su un gestore code inattivo, utilizzando il comando:

```
endmqclsr [-m QMNAME]
```

Se non si specifica un nome gestore code, viene utilizzato il gestore code predefinito.

Linux AIX Utilizzo dell'opzione TCP/IP SO_KEEPALIVE

Su alcuni sistemi AIX and Linux , è possibile definire il tempo di attesa TCP prima di verificare che la connessione sia ancora disponibile e la frequenza con cui tenta nuovamente la connessione se il primo controllo ha esito negativo. Questo è un parametro ottimizzabile del kernel o può essere immesso sulla riga comandi.

Se si desidera utilizzare l'opzione SO_KEEPALIVE (per ulteriori informazioni, consultare [“Verifica che l'altra estremità del canale sia ancora disponibile”](#) a pagina 242) è necessario aggiungere la seguente voce al file di configurazione del gestore code (qm.ini):

```
TCP:
KeepAlive=yes
```

Consultare la documentazione per il sistema AIX o Linux per ulteriori informazioni.

Linux AIX Definizione di una connessione LU 6.2 su AIX and Linux

SNA deve essere configurato in modo che sia possibile stabilire una conversazione LU 6.2 tra le macchine.

Per le informazioni più recenti sulla configurazione SNA su TCP/IP, consultare la seguente documentazione in linea IBM : [Communications Server](#).

SNA deve essere configurato in modo che sia possibile stabilire una conversazione LU 6.2 tra i sistemi.

Per informazioni, consultare il manuale *Multiplatform APPC Configuration Guide* e la seguente tabella.

Piattaforma remota	TPNAME	TPPATH
z/OS senza CICS	Lo stesso del TPName corrispondente nelle informazioni laterali sul gestore code remoto.	-
z/OS utilizzo CICS	CKRC (mittente) CKSV (richiedente) CKRC (server)	-
IBM i	Uguale al valore di confronto nella specifica di instradamento sul sistema IBM i .	-

Tabella 25. Impostazioni sul sistema AIX o Linux locale per una piattaforma del gestore code remoto (Continua)

Piattaforma remota	TPNAME	TPPATH
Sistemi AIX and Linux	Lo stesso del TPName corrispondente nelle informazioni laterali sul gestore code remoto.	<code>MQ_INSTALLATION_PATH/bin/amqcrs6a</code>
Windows	Come specificato nel comando Windows Esegui listener o nel programma di transazione richiamabile definito utilizzando TpSetup su Windows.	<code>MQ_INSTALLATION_PATH\bin\amqcrs6a</code>

`MQ_INSTALLATION_PATH` rappresenta la directory di livello superiore in cui è installato IBM MQ .

Se si dispone di più di un gestore code sulla stessa macchina, verificare che i nomi TP nelle definizioni di canale siano univoci.

Concetti correlati

[“Invio fine su LU 6.2 su AIX and Linux” a pagina 281](#)

Su sistemi AIX and Linux , creare un oggetto lato CPI-C (destinazione simbolica) e immettere questo nome nel campo Nome connessione nella definizione del canale. Creare anche un collegamento LU 6.2 al partner.

[“Ricezione su LU 6.2 su AIX and Linux” a pagina 281](#)

Sui sistemi AIX and Linux , creare un collegamento di ascolto all'estremità ricevente, un profilo di connessione logica LU 6.2 e un profilo TPN.

Linux **AIX** *Invio fine su LU 6.2 su AIX and Linux*

Su sistemi AIX and Linux , creare un oggetto lato CPI-C (destinazione simbolica) e immettere questo nome nel campo Nome connessione nella definizione del canale. Creare anche un collegamento LU 6.2 al partner.

Nell'oggetto lato CPI-C, immettere il nome LU partner sulla macchina ricevente, il nome del programma di transazione e il nome del modo. Ad esempio:

```
Partner LU Name          REMHOST
Remote TP Name          recv
Service Transaction Program no
Mode Name                #INTER
```

Viene utilizzato SECURITY PROGRAM, dove supportato da CPI-C, quando IBM MQ tenta di stabilire una sessione SNA.

Linux **AIX** *Ricezione su LU 6.2 su AIX and Linux*

Sui sistemi AIX and Linux , creare un collegamento di ascolto all'estremità ricevente, un profilo di connessione logica LU 6.2 e un profilo TPN.

Nel profilo TPN, immettere il percorso completo del file eseguibile e il nome del programma di transazione:

```
Full path to TPN executable  MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name    recv
User ID                      0
```

`MQ_INSTALLATION_PATH` rappresenta la directory di livello superiore in cui è installato IBM MQ .

Sui sistemi in cui è possibile impostare l'ID utente, specificare un utente che sia un membro del gruppo mqm.

AIX

Su AIX, impostare le variabili di ambiente APPCTPN (nome transazione) e APPCLLU (nome LU locale) (è possibile utilizzare i pannelli di configurazione per il programma di transazione richiamato).

Potrebbe essere necessario utilizzare un gestore code diverso da quello predefinito. In tal caso, definire un file di comandi che richiama:

```
amqcrs6a -m Queue_Man_Name
```

quindi richiamare il file di comandi.

IBM i

Monitoraggio e controllo dei canali su IBM i

Utilizzare i comandi e i pannelli DQM per creare, monitorare e controllare i canali dei gestori code remoti. Ogni gestore code ha un programma DQM per il controllo delle interconnessioni ai gestori code remoti compatibili.

Informazioni su questa attività

Il seguente elenco è una breve descrizione dei componenti della funzione di controllo del canale:

- Le definizioni di canale vengono conservate come oggetti del gestore code.
- I comandi del canale sono una sottoserie della serie di comandi IBM MQ for IBM i .

Utilizzare il comando GO CMDMQM per visualizzare la serie completa di comandi IBM MQ for IBM i .

- Utilizzare i pannelli di definizione del canale o i comandi per:
 - Creare, copiare, visualizzare, modificare ed eliminare le definizioni di canale
 - Avviare e arrestare i canali, eseguire il ping, ripristinare i numeri di sequenza dei canali e risolvere i messaggi in dubbio quando i collegamenti non possono essere ristabiliti
 - Visualizza informazioni di stato sui canali
- I canali possono essere gestiti anche utilizzando MQSC
- I canali possono essere gestiti anche utilizzando IBM MQ Explorer.
- I numeri di sequenza e gli identificativi *LUW* (*logical unit of work*) vengono memorizzati nel file di sincronizzazione e utilizzati per la sincronizzazione del canale.

È possibile utilizzare i comandi e i pannelli per: definire i canali dei messaggi e gli oggetti associati e monitorare e controllare i canali dei messaggi. Utilizzando il pulsante F4=Prompt , è possibile specificare il relativo gestore code. Se non si utilizza il prompt, viene utilizzato il gestore code predefinito. Con F4=Prompt, viene visualizzato un pannello aggiuntivo in cui è possibile immettere il nome del gestore code pertinente e, a volte, altri dati.

Gli oggetti da definire con i pannelli sono:

- Code di trasmissione
- Definizioni di coda remota
- Definizioni alias gestore code
- Definizioni alias coda di risposta
- Code locali di risposta
- Definizioni di canali di messaggi

Per ulteriori informazioni sui concetti coinvolti nell'utilizzo di questi oggetti, consultare [“Configurazione dell'accodamento distribuito” a pagina 205.](#)

I canali devono essere completamente definiti e i relativi oggetti associati devono esistere ed essere disponibili per l'utilizzo, prima che un canale possa essere avviato.

Inoltre, il link di comunicazione particolare per ogni canale deve essere definito e disponibile prima che un canale possa essere eseguito. Per una descrizione della modalità di definizione dei collegamenti LU 6.2 e TCP/IP, consultare la guida alla comunicazione specifica per l'installazione.

Procedura

- Per ulteriori informazioni sulla creazione e l'utilizzo degli oggetti, consultare:
 - [“Creazione di oggetti su IBM i” a pagina 283](#)
 - [“Creazione di un canale su IBM i” a pagina 283](#)
 - [“Avvio di un canale su IBM i” a pagina 285](#)
 - [“Selezione di un canale su IBM i” a pagina 286](#)
 - [“Esplorazione di un canale su IBM i” a pagina 286](#)
 - [“Ridenominazione di un canale su IBM i” a pagina 288](#)
 - [“Gestione dello stato del canale su IBM i” a pagina 288](#)
 - [“Opzioni di gestione canale su IBM i” a pagina 289](#)

Concetti correlati

[“Impostazione della comunicazione per IBM i” a pagina 295](#)

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché abbia esito positivo, è necessario che la connessione sia definita e disponibile.

Attività correlate

[“Configurazione delle connessioni tra client e server” a pagina 16](#)

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

Riferimenti correlati

[Configurazione di esempio - IBM MQ for IBM i](#)

[Esempio di pianificazione del canale di messaggi per IBM MQ for IBM i](#)

[Comandi CL di IBM MQ for IBM i](#)

Creazione di oggetti su IBM i

È possibile utilizzare il comando CRTMQMQ per creare la coda e gli oggetti alias.

È possibile creare gli oggetti coda e alias, come ad esempio: code di trasmissione, definizioni di code remote, definizioni di alias del gestore code, definizioni di alias della coda reply - to e code locali reply - to.

Per un elenco di oggetti predefiniti, vedere [Sistema e oggetti predefiniti](#).

Creazione di un canale su IBM i

È possibile creare un canale dal pannello Crea canale o utilizzando il comando CRTMQMCHL sulla riga comandi.

Per creare un canale:

1. Utilizzare F6 dal pannello Gestione canali MQM (WRKMQMCHL).
 - In alternativa, utilizzare il comando CRTMQMCHL dalla riga comandi.
 - In entrambi i casi, viene visualizzato il pannello Crea canale. Tipo:
 - Il nome del canale nel campo fornito
 - Il tipo di canale per questa estremità del link
2. Premere Invio.

Nota: È necessario denominare tutti i canali nella rete in modo univoco. Come mostrato in [Diagramma di rete che mostra tutti i canali](#), inclusi i nomi dei gestori code di origine e di destinazione nel nome del canale, è un buon modo per farlo.

Le voci vengono convalidate e gli errori vengono riportati immediatamente. Correggere gli errori e continuare.

Viene visualizzato il pannello delle impostazioni del canale appropriato per il tipo di canale scelto. Completare i campi con le informazioni precedentemente raccolte. Premere Invio per creare il canale.

Viene fornita una guida per decidere il contenuto dei vari campi nelle descrizioni dei pannelli di definizione del canale nei pannelli di aiuto e in [Attributi canale](#).

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . > CHANNAME_____
Channel type . . . . . > *SDR__   *RCVR, *SDR, *SVR, *RQSTR...
Message Queue Manager name      *DFT_____

Replace . . . . . *NO_   *NO, *YES
Transport type . . . . . *TCP_____ *LU62, *TCP, *SYSDFTCHL
Text 'description' . . . . . > 'Example Channel Definition'_____

Connection name . . . . . *SYSDFTCHL_____

-----
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Figura 25. Crea canale (1)

```
Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Transmission queue . . . . . 'TRANSMISSION_QUEUE_NAME'_____

-----
Message channel agent . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
Message channel agent user ID . . *SYSDFTCHL__ Character value...
Coded Character Set Identifier *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Batch size . . . . . 50_____ 1-9999, *SYSDFTCHL
Disconnect interval . . . . . 6000_____ 1-999999, *SYSDFTCHL
Short retry interval . . . . . 60_____ 0-999999999, *SYSDFTCHL
Short retry count . . . . . 10_____ 0-999999999, *SYSDFTCHL
Long retry interval . . . . . 1200_____ 0-999999999, *SYSDFTCHL
Long retry count . . . . . 999999999__ 0-999999999, *SYSDFTCHL
Security exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
Security exit user data . . . . . *SYSDFTCHL_____

-----
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Figura 26. Crea canale (2)

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Send exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values
Send exit user data . . . . . _____
+ for more values
Receive exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values
-----
Receive exit user data . . . . . _____
+ for more values
Message exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values
-----
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

Figura 27. Crea canale (3)

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Message exit user data . . . . . _____
+ for more values
Convert message . . . . . *SYSDFTCHL_ *YES, *NO, *SYSDFTCHL
Sequence number wrap . . . . . 99999999__ 100-99999999, *SYSDFTCHL
Maximum message length . . . . . 4194304___ 0-4194304, *SYSDFTCHL
Heartbeat interval . . . . . 300_____ 0-999999999, *SYSDFTCHL
Non Persistent Message Speed . . *FAST_____ *FAST, *NORMAL, *SYSDFTCHL
Password . . . . . *SYSDFTCHL_ Character value, *BLANK...
Task User Profile . . . . . *SYSDFTCHL_ Character value, *BLANK...
Transaction Program Name . . . . *SYSDFTCHL

```

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figura 28. Crea canale (4)

Avvio di un canale su IBM i

È possibile avviare un canale dal pannello Gestione canali oppure utilizzando il comando STRMQMCHL sulla riga comandi.

I listener sono validi solo per TCP. Per i listener SNA, è necessario configurare il sottosistema di comunicazione.

Per consentire alle applicazioni di scambiare messaggi, è necessario avviare un programma listener per le connessioni in entrata utilizzando il comando STRMQMLSR.

Per connessioni in uscita, è necessario avviare il canale in uno dei modi seguenti:

1. Utilizzare il comando CL STRMQMCHL, specificando il nome del canale, per avviare il canale come un processo o un thread, in base al parametro MCATYPE. (Se i canali vengono avviati come thread, sono thread di un iniziatore di canali.)

```
STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
```

2. Utilizzare un iniziatore di canali per attivare il canale. Un iniziatore di canali viene avviato automaticamente all'avvio del gestore code. Questo avvio automatico può essere eliminato modificando la stanza chinit nel file qm.ini per tale gestore code.
3. Utilizzare il comando WRKMQMCHL per avviare il pannello Gestione canali e scegliere l'opzione 14 per avviare un canale.

IBM i Selezione di un canale su IBM i

È possibile selezionare un canale dal pannello Gestione canali.

Per selezionare un canale, utilizzare il comando WRKMQMCHL per iniziare dal pannello Gestione canali:

1. Spostare il cursore sul campo dell'opzione associato al nome del canale richiesto.
2. Immettere un numero di opzione.
3. Premere Invio per attivare la scelta.

Se si seleziona più di un canale, le opzioni vengono attivate in sequenza.

```
Work with MQM Channels

Queue Manager Name . . : CNX

Type options, press Enter.
2=Change 3=Copy 4=Delete 5=Display 8=Work with Status 13=Ping
14=Start 15=End 16=Reset 17=Resolve

Opt  Name          Type      Transport  Status
CHLNIC          *RCVR    *TCP       INACTIVE
CORSAIR.TO.MUSTANG *SDR     *LU62      INACTIVE
FV.CHANNEL.MC.DJE1 *RCVR    *TCP       INACTIVE
FV.CHANNEL.MC.DJE2 *SDR     *TCP       INACTIVE
FV.CHANNEL.MC.DJE3 *RQSTR   *TCP       INACTIVE
FV.CHANNEL.MC.DJE4 *SVR     *TCP       INACTIVE
FV.CHANNEL.PETER  *RCVR    *TCP       INACTIVE
FV.CHANNEL.PETER.LU *RCVR    *LU62      INACTIVE
FV.CHANNEL.PETER.LU1 *RCVR    *LU62      INACTIVE
More...
Parameters or command
===>
F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F12=Cancel
F21=Print
```

Figura 29. Gestione canali

IBM i Esplorazione di un canale su IBM i

È possibile sfogliare un canale dal pannello Visualizza canale o utilizzando il comando DSPMQMCHL sulla riga comandi.

Per visualizzare le impostazioni di un canale, utilizzare il comando WRKMQMCHL per iniziare dal pannello Visualizzazione canale:

1. Immettere l'opzione 5 (Visualizzazione) rispetto al nome canale richiesto.
2. Premere Invio per attivare la scelta.

Se si seleziona più di un canale, vengono presentati in sequenza.

In alternativa, è possibile utilizzare il comando DSPMQMCHL dalla riga comandi.

Ciò comporta la visualizzazione del pannello Visualizzazione canale appropriato con i dettagli delle impostazioni correnti per il canale. I campi sono descritti in [Attributi canale](#).

```

Display MQM Channel

Channel name . . . . . : ST.JST.2T01
Queue Manager Name . . . . . : QMREL
Channel type . . . . . : *SDR
Transport type . . . . . : *TCP
Text 'description' . . . . . : John's sender to WINSDOA1

Connection name . . . . . : MUSTANG

Transmission queue . . . . . : WINSDOA1

Message channel agent . . . . . :
Library . . . . . :
Message channel agent user ID : *NONE
Batch interval . . . . . : 0
Batch size . . . . . : 50
Disconnect interval . . . . . : 6000

F3=Exit F12=Cancel F21=Print

```

Figura 30. Visualizzazione di un canale TCP/IP (1)

```

Display MQM Channel

Short retry interval . . . . . : 60
Short retry count . . . . . : 10
Long retry interval . . . . . : 6000
Long retry count . . . . . : 10
Security exit . . . . . :
Library . . . . . :
Security exit user data . . . . . :
Send exit . . . . . :
Library . . . . . :
Send exit user data . . . . . :
Receive exit . . . . . :
Library . . . . . :
Receive exit user data . . . . . :
Message exit . . . . . :
Library . . . . . :
Message exit user data . . . . . :
More...

F3=Exit F12=Cancel F21=Print

```

Figura 31. Visualizzazione di un canale TCP/IP (2)

```
Display MQM Channel
Sequence number wrap . . . . . : 999999999
Maximum message length . . . . : 10000
Convert message . . . . . : *NO
Heartbeat interval . . . . . : 300
Nonpersistent message speed . . *FAST
```

Bottom

F3=Exit F12=Cancel F21=Print

Figura 32. Visualizzare un canale TCP/IP (3)

IBM i Ridenominazione di un canale su IBM i

È possibile ridenominare un canale dal pannello Gestione canali.

Per ridenominare un canale di messaggi, iniziare dal pannello Gestione canali:

1. Chiudere il canale.
2. Utilizzare l'opzione 3 (Copia) per creare un duplicato con il nuovo nome.
3. Utilizzare l'opzione 5 (Visualizzazione) per verificare che sia stata creata correttamente.
4. Utilizzare l'opzione 4 (Cancellazione) per eliminare il canale originale.

Se si decide di rinominare un canale di messaggi, assicurarsi che entrambe le estremità del canale vengano rinominate contemporaneamente.

IBM i Gestione dello stato del canale su IBM i

È possibile gestire lo stato del canale dal pannello Gestione stato del canale.

Utilizzare il comando WRKMQMCHST per visualizzare il primo di una serie di pannelli che mostrano lo stato dei canali. È possibile visualizzare i pannelli di stato in sequenza quando si seleziona Modifica vista (F11).

In alternativa, selezionando l'opzione 8 (Gestisci stato) dal pannello Gestisci canali MQM viene visualizzato anche il primo pannello di stato.

MQSeries Work with Channel Status

Type options, press Enter.

5=Display 13=Ping 14=Start 15=End 16=Reset 17=Resolve

Opt Name	Connection	Indoubt	Last Seq
CARTS_CORSAIR_CHAN	GBIBMIYA.WINSDOA1	NO	1
CHLNIC	9.20.2.213	NO	3
FV.CHANNEL.PETER2	9.20.2.213	NO	6225
JST.1.2	9.20.2.201	NO	28
MP_MUST_TO_CORS	9.20.2.213	NO	100
MUSTANG.TO.CORSAIR	GBIBMIYA.WINSDOA1	NO	10
MP_CORS_TO_MUST	9.20.2.213	NO	101
JST.2.3	9.5.7.126	NO	32
PF_WINSDOA1_LU62	GBIBMIYA.IYA80020	NO	54
PF_WINSDOA1_LU62	GBIBMIYA.WINSDOA1	NO	500
ST.JCW.EXIT.2T01.CHL	9.20.2.213	NO	216

Bottom

Parameters or command

==>

F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F11=Change view

F12=Cancel F21=Print

Figura 33. Primo della serie di pannelli di stato del canale

Le opzioni disponibili nel pannello Gestione stato canale sono:

Opzione di menu	Descrizione
5=Display	Visualizza le impostazioni del canale.
13=Ping	Avvia un'azione Ping, dove appropriato.
14=Start	Avvia il canale.
15=End	Arresta il canale.
16=Reset	Reimposta il numero di sequenza del canale.
17=Resolve	Risolve manualmente una situazione di canale in dubbio.

IBM i Opzioni di gestione canale su IBM i

Il pannello Gestione canali viene raggiunto con il comando WRKMQMCHL e consente di monitorare lo stato di tutti i canali elencati e di emettere comandi per canali selezionati.

Le opzioni disponibili nel pannello Gestione canale sono:

Opzione di menu	Descrizione
<u>"2=Change" a pagina 290</u>	Modifica gli attributi di un canale.
<u>"3=Copy" a pagina 290</u>	Copia gli attributi di un canale in un nuovo canale.
<u>"4=Delete" a pagina 290</u>	Elimina un canale.
<u>"5=Display" a pagina 290</u>	Visualizza le impostazioni correnti per il canale.
<u>"6=Create" a pagina 290</u>	Visualizza il pannello Crea canale
<u>"8=Work con stato" a pagina 291</u>	Visualizza i pannelli di stato del canale.
<u>"13=Ping" a pagina 292</u>	Esegue la funzione Ping per verificare la connessione al sistema adiacente scambiando un messaggio di dati fisso con l'estremità remota.

Opzione di menu	Descrizione
“14=Start” a pagina 292	Avvia il canale selezionato o reimposta un canale ricevente disabilitato.
“15=End” a pagina 293	Richiede la chiusura del canale.
“16=Reset” a pagina 294	Richiede al canale di reimpostare i numeri di sequenza su questa estremità del collegamento. I numeri devono essere uguali ad entrambe le estremità per l'avvio del canale.
“17=Resolve” a pagina 294	Richiede al canale di risolvere i messaggi in dubbio senza stabilire una connessione all'altra estremità.
“18=Visualizza autorizzazione” a pagina 294	Visualizza l'autorizzazione dell'oggetto IBM MQ
“19=Concedi autorizzazione” a pagina 294	Concede l'autorizzazione all'oggetto IBM MQ
“20=Revoca autorizzazione” a pagina 295	Revoca l'autorizzazione sull'oggetto IBM MQ
“21=Ripristina oggetto” a pagina 295	Recupera l'oggetto IBM MQ
“22=Registrazione immagine” a pagina 295	Immagine oggetto Record IBM MQ

IBM i 2=Change

Utilizzare l'opzione Modifica per modificare una definizione di canale esistente.

L'opzione Modifica, o il comando CHGMQMCHL, modifica una definizione di canale esistente, ad eccezione del nome del canale. Sovrascrivere i campi da modificare nel pannello di definizione del canale e salvare la definizione aggiornata premendo Invio.

IBM i 3=Copy

Utilizzare l'opzione Copia per copiare un canale esistente.

L'opzione Copia utilizza il comando CPYMQMCHL per copiare un canale esistente. Il pannello Copia consente di definire il nome del nuovo canale. Tuttavia, è necessario limitare i caratteri utilizzati ai caratteri validi per i nomi oggetto IBM i ; consultare [Amministrazione IBM MQ for IBM i](#).

Premere Invio sul pannello Copia per visualizzare i dettagli delle impostazioni correnti. È possibile modificare qualsiasi nuova impostazione del canale. Salvare la nuova definizione di canale premendo Invio.

IBM i 4>Delete

Utilizzare l'opzione Elimina per eliminare il canale selezionato.

Viene visualizzato un pannello per confermare o annullare la richiesta.

IBM i 5=Display

Utilizzare l'opzione Visualizzazione per visualizzare le definizioni correnti per il canale.

Questa scelta visualizza il pannello con i campi che mostrano i valori correnti dei parametri e protetti dall'input utente.

IBM i 6=Create

Utilizzare l'opzione Crea per visualizzare il pannello Crea canale.

Utilizzare l'opzione Crea oppure immettere il comando CRTMQMCHL dalla riga comandi per ottenere il pannello Crea canale. Esistono esempi di pannelli Crea canale, a partire da [Figura 25 a pagina 284](#).

Con questo pannello, si crea una definizione di canale da una schermata di campi riempiti con valori predefiniti forniti da IBM MQ for IBM i. Immettere il nome del canale, selezionare il tipo di canale che si sta creando e il metodo di comunicazione da utilizzare.

Quando si preme Invio, viene visualizzato il pannello. Immettere le informazioni in tutti i campi richiesti in questo pannello e nei pannelli rimanenti, quindi salvare la definizione premendo Invio.

Il nome del canale deve essere lo stesso ad entrambe le estremità del canale e univoco nella rete. Tuttavia, è necessario limitare i caratteri utilizzati ai caratteri validi per nomi oggetto IBM MQ for IBM i.

Tutti i pannelli hanno valori predefiniti forniti da IBM MQ for IBM i per alcuni campi. È possibile personalizzare questi valori oppure modificarli durante la creazione o la copia dei canali. Per personalizzare i valori, consultare *IBM MQ for IBM i Amministrazione di sistema*.

È possibile creare la propria serie di valori predefiniti del canale impostando canali dummy con i valori predefiniti richiesti per ciascun tipo di canale e copiandoli ogni volta che si desidera creare nuove definizioni di canale.

Riferimenti correlati

[Attributi canale](#)

8=Work con stato

Utilizzare Gestione stato per visualizzare informazioni dettagliate sullo stato del canale.

La colonna di stato indica se il canale è attivo o inattivo e viene visualizzato continuamente nel pannello Gestisci canali MQM. Usare l'opzione 8 (Gestione stato) per visualizzare ulteriori informazioni sullo stato. In alternativa, queste informazioni possono essere visualizzate dalla riga comandi con il comando WRKMQMCHST. Consultare [“Gestione dello stato del canale su IBM i” a pagina 288](#).

- Nome canale
- Tipo di canale
- Stato dei canali
- Istanza canale
- Gestore code remoto
- Nome coda di trasmissione
- Nome connessione di comunicazione
- Stato in dubbio del canale
- Ultimo numero sequenza
- Numero di messaggi in dubbio
- Sequenza in sospeso
- Numero di messaggi nella coda di trasmissione
- Identificativo LUW (Logical unit of work)
- Identificativo LUW (logical unit of work / unità logica di lavoro) in dubbio
- Stato secondario canale
- Controllo canale
- Compressione intestazione
- Compressione messaggi
- Indicatore tempo di compressione
- Indicatore di velocità di compressione
- Indicatore tempo di coda della trasmissione
- Indicatore tempo di rete

- Indicatore ora di uscita
- Indicatore dimensione batch
- Conversazioni condivise correnti
- Numero massimo di conversazioni condivise

IBM i **13=Ping**

Utilizzare l'opzione Ping per scambiare un messaggio di dati fisso con l'estremità remota.

Un ping IBM MQ eseguito correttamente fornisce al supervisore del sistema la certezza che il canale sia disponibile e funzionante.

Il ping non comporta l'utilizzo di code di trasmissione e code di destinazione. Utilizza le definizioni di canale, il link di comunicazione correlato e l'impostazione di rete.

È disponibile solo dai canali mittente e server. Il canale corrispondente viene avviato sul lato lontano del link ed esegue la negoziazione del parametro di avvio. Gli errori vengono notificati normalmente.

Il risultato dello scambio di messaggi viene presentato nel pannello Ping ed è il testo del messaggio restituito, insieme all'ora in cui è stato inviato il messaggio e all'ora in cui è stata ricevuta la risposta.

Ping con LU 6.2

Quando il Ping viene richiamato in IBM MQ for IBM i, viene eseguito con l>ID utente dell'utente che richiede la funzione, mentre il normale modo in cui viene eseguito un programma del canale è per l>ID utente QMQM da utilizzare per i programmi del canale. L>ID utente passa al lato ricevente e deve essere valido sul lato ricevente per l'assegnazione della conversazione LU 6.2 .

IBM i **14=Start**

Utilizzare l'opzione Avvia per avviare un canale manualmente.

L'opzione di avvio è disponibile per i canali mittente, server e richiedente. Non è necessario dove è stato impostato un canale con il trigger del gestore code.

L'opzione Start viene utilizzata anche per i canali ricevente, server - connessione, mittente cluster e ricevente cluster. L'avvio di un canale ricevente in stato STOPPED indica che può essere avviato dal canale remoto.

Quando avviato, l'MCA mittente legge il file di definizione del canale e apre la coda di trasmissione. Viene emessa una sequenza di avvio del canale, che avvia in remoto l'MCA corrispondente del canale ricevente o server. Una volta avviati, i processi del mittente e del server attendono i messaggi in arrivo sulla coda di trasmissione e li trasmettono non appena arrivano.

Quando si utilizza il trigger, è necessario avviare il processo trigger in esecuzione continua per monitorare la coda di iniziazione. Il comando STRMQMCHLI può essere utilizzato per avviare il processo.

All'estremità di un canale, il processo di ricezione potrebbe essere avviato in risposta all'avvio di un canale dall'estremità di invio. Il metodo per eseguire questa operazione è diverso per i canali connessi LU 6.2 e TCP/IP:

- I canali connessi LU 6.2 non richiedono alcuna azione esplicita all'estremità di ricezione di un canale.
- I canali TCP connessi richiedono l'esecuzione continua di un processo listener. Questo processo attende le richieste di avvio del canale dall'estremità remota del collegamento e avvia il processo definito nelle definizioni di canale per tale connessione.

Quando il sistema remoto è IBM i, è possibile utilizzare il comando STRMQMLSR.

L'utilizzo dell'opzione Start provoca sempre la risincronizzazione del canale, se necessario.

Perché l'inizio abbia successo:

- Le definizioni di canale, locale e remoto devono esistere. Se non esiste una definizione di canale appropriata per un canale ricevente o di connessione server, ne viene creata automaticamente una predefinita se il canale è definito automaticamente. Vedere [Programma di uscita di definizione automatica del canale](#).
- La coda di trasmissione deve esistere, essere abilitata per GET e non avere altri canali che la utilizzano.
- Gli MCA, locali e remoti, devono esistere.
- Il link di comunicazione deve essere disponibile.
- I gestori code devono essere in esecuzione, locali e remoti.
- Il canale messaggi deve essere inattivo.

Per trasferire i messaggi, devono esistere le code remote e le definizioni di coda remota.

Viene restituito un messaggio al pannello che conferma che la richiesta di avviare un canale è stata accettata. Per confermare che il processo di avvio ha avuto esito positivo, controllare il log di sistema o premere F5 (aggiornare il pannello).

15=End

Utilizzare Fine per arrestare l'attività del canale

Utilizzare l'opzione Fine per richiedere al canale di arrestare l'attività. Il canale non invia ulteriori messaggi.

Selezionare F4 prima di premere Invio per scegliere se il canale diventa ARRESTATO o INATTIVO e se arrestare il canale utilizzando un arresto CONTROLLATO o IMMEDIATE. Un canale arrestato deve essere riavviato dall'operatore per diventare nuovamente attivo. È possibile attivare un canale inattivo.

Arresta immediatamente

Utilizzare Arresta immediatamente per arrestare un canale senza completare alcuna unità di lavoro.

Questa opzione termina il processo del canale. Di conseguenza, il canale non completa l'elaborazione del batch di messaggi corrente e, pertanto, non può lasciare il canale in dubbio. In generale, è meglio per gli operatori utilizzare l'opzione di arresto controllato.



Arresto controllato

Utilizzare Arresta controllato per arrestare un canale alla fine dell'unità di lavoro corrente.

Questa scelta richiede la chiusura ordinata del canale; il batch di messaggi corrente viene completato e la procedura del punto di sincronizzazione viene eseguita con l'altra estremità del canale.

Riavvio dei canali arrestati

Quando un canale passa allo stato ARRESTATO, è necessario riavviarlo manualmente. Puoi riavviare il canale nei modi seguenti:

- Utilizzando il comando MQSC **START CHANNEL** .
- Utilizzando il comando PCF **Start Channel** .
- Utilizzando IBM MQ Explorer.
-  Su z/OS, utilizzando il pannello Avvia un canale.
-  Su IBM i, utilizzando il comando **STRMQMCHL CL** o l'opzione **START** sul pannello WRKMQMCHL.

Per i canali mittente o server, quando il canale è entrato nello stato STOPPED, la coda di trasmissione associata è stata impostata su GET (DISABLED) e il trigger è stato disattivato. Quando viene ricevuta la richiesta di avvio, questi attributi vengono reimpostati automaticamente.

z/OS Se l'iniziatore del canale si arresta mentre un canale si trova nello stato REENTAMENTO o ARRESTATO, lo stato del canale viene ricordato quando l'iniziatore del canale viene riavviato. Tuttavia, lo stato del canale per il tipo di canale SVRCONN viene reimpostato se l'iniziatore del canale si arresta mentre il canale è in stato ARRESTATO.

Multi Se il gestore code si arresta mentre un canale si trova nello stato REENTAMENTO o ARRESTATO, lo stato del canale viene ricordato quando il gestore code viene riavviato. Da IBM MQ 8.0 in poi, ciò si applica anche ai canali SVRCONN. In precedenza, lo stato del canale per il tipo di canale SVRCONN era stato reimpostato se l'iniziatore del canale era stato arrestato mentre il canale era in stato ARRESTATO.

IBM i **16=Reset**

Utilizzare l'opzione Reimposta per forzare una nuova sequenza di messaggi.

L'opzione Reimposta modifica il numero di sequenza del messaggio. Utilizzarlo con attenzione e solo dopo aver utilizzato l'opzione Risolvi per risolvere eventuali situazioni in dubbio. Questa opzione è disponibile solo sul canale mittente o server. Il primo messaggio avvia la nuova sequenza al successivo avvio del canale.

IBM i **17=Resolve**

Utilizzare l'opzione Risolvi per forzare un commit locale o un backout dei messaggi in dubbio conservati in una coda di trasmissione.

Utilizzare l'opzione Risolvi quando i messaggi sono in dubbio da parte di un mittente o di un server, ad esempio perché un'estremità del collegamento è terminata e non vi è alcuna possibilità che venga ripristinata. L'opzione Resolve accetta uno dei seguenti due parametri: BACKOUT o COMMIT. Il backout ripristina i messaggi nella coda di trasmissione, mentre il commit li elimina.

Il programma del canale non tenta di stabilire una sessione con un partner. Invece, determina l'identificativo LUWID (logical unit of work identifier) che rappresenta i messaggi in dubbio. Quindi, come richiesto, emette:

- BACKOUT per ripristinare i messaggi nella coda di trasmissione; oppure
- COMMIT per cancellare i messaggi dalla coda di trasmissione.

Affinché la risoluzione abbia successo:

- Il canale deve essere inattivo
- Il canale deve essere in dubbio
- Il tipo di canale deve essere mittente o server
- La definizione del canale, locale, deve esistere
- Il gestore code deve essere in esecuzione, locale

IBM i **18=Visualizza autorizzazione**

Utilizzare l'opzione Visualizzazione autorizzazione per visualizzare le azioni che un utente è autorizzato ad eseguire su un determinato oggetto IBM MQ .

Per un oggetto scelto e per un utente, il comando DSPMQAUT mostra le autorizzazioni di cui l'utente dispone per eseguire azioni su un oggetto IBM MQ . Se l'utente è un membro di più gruppi, il comando mostra l'autorizzazione combinata di tutti i gruppi all'oggetto.

IBM i **19=Concedi autorizzazione**

Utilizzare l'opzione Concedi autorizzazione per concedere l'autorità di eseguire azioni sugli oggetti IBM MQ a un altro utente o gruppo di utenti.

il comando GRMQMAUT è disponibile solo per gli utenti del gruppo QMQMADM. Un utente in QMQMADM concede l'autorizzazione ad altri utenti per eseguire azioni sugli oggetti IBM MQ indicati nel comando identificando gli utenti per nome o concedendo l'autorizzazione a tutti gli utenti in *PUBLIC.

IBM i **20=Revoca autorizzazione**

Utilizzare Revoca autorizzazione per eliminare l'autorizzazione ad eseguire azioni sugli oggetti dagli utenti.

Il comando RVKMQMAUT è disponibile solo per gli utenti del gruppo QMQMADM. Un utente nel gruppo QMQMADM rimuove l'autorizzazione da altri utenti per eseguire azioni sugli oggetti IBM MQ indicati nel comando identificando gli utenti per nome o revocando l'autorizzazione a tutti gli utenti in *PUBLIC.

IBM i **21=Ripristina oggetto**

Utilizzare Ripristina oggetto per ripristinare gli oggetti danneggiati dalle informazioni memorizzate nei giornali IBM MQ .

Recupera oggetto utilizza il comando RCRMQMOBJ (Re-create MQ Object) per recuperare tutti gli oggetti danneggiati indicati nel comando. Se un oggetto non è danneggiato, non viene eseguita alcuna azione su tale oggetto.

IBM i **22=Registrazione immagine**

Utilizzare l'immagine record per ridurre il numero di ricevitori di giornale richiesti per il recupero di una serie di oggetti e per ridurre al minimo il tempo di recupero.

Il comando RCDMQMIMG prende un punto di controllo per tutti gli oggetti selezionati nel comando. Sincronizza i valori correnti degli oggetti nell'IFS (integrated file system) con le informazioni successive sugli oggetti, come MQPUT e MQGET registrati nei ricevitori di giornale.

Quando il comando viene completato, gli oggetti nell'IFS sono aggiornati e non è più necessario che tali ricevitori di giornale siano presenti per ripristinare gli oggetti. Tutti i ricevitori di giornale scollegati possono essere scollegati (purché non sia necessario che siano presenti per ripristinare altri oggetti).

IBM i **Impostazione della comunicazione per IBM i**

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché abbia esito positivo, è necessario che la connessione sia definita e disponibile.

DQM è una funzione di accodamento remoto per IBM MQ for IBM i. Fornisce programmi di controllo del canale per il Gestore code IBM MQ for IBM i che formano l'interfaccia per i collegamenti di comunicazione, controllabili dall'operatore di sistema.

Quando un canale di gestione dell'accodamento distribuito viene avviato, tenta di utilizzare la connessione specificata nella definizione del canale. Perché abbia esito positivo, è necessario che la connessione sia definita e disponibile. Questa sezione spiega come verificare che la connessione sia definita e disponibile.

Prima di avviare un canale, la coda di trasmissione deve essere definita come descritto in questa sezione e deve essere inclusa nella definizione del canale di messaggi.

È possibile scegliere l'etere delle seguenti due forme di comunicazione tra sistemi IBM MQ for IBM i :

- [“Definizione di un collegamento TCP su IBM i” a pagina 296](#)

Per TCP, è possibile utilizzare un indirizzo host e tali connessioni sono configurate come descritto in *IBM i Communication Configuration Reference*.

Nell'ambiente TCP, a ogni servizio distribuito viene assegnato un unico indirizzo TCP che può essere utilizzato dalle macchine remote per accedere al servizio. L'indirizzo TCP è composto da un nome host / numero e un numero di porta. Tutti i gestori code utilizzano tale numero per comunicare tra loro tramite TCP.

- [“Ricezione su TCP” a pagina 297](#)

Questa forma di comunicazione richiede la definizione di una IBM i unità logica SNA di tipo 6.2 (LU 6.2) che fornisce il collegamento fisico tra il sistema IBM i che serve il gestore code locale e quello che serve il gestore code remoto. Fare riferimento al manuale *IBM i Communication Configuration Reference* per dettagli sulla configurazione delle comunicazioni in IBM i.

Ove necessario, l'accordo di attivazione deve essere preparato anche con la definizione dei processi e delle code necessari.

MQ Adv. **CD** Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linuxo Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Attività correlate

“Monitoraggio e controllo dei canali su IBM i” a pagina 282

Utilizzare i comandi e i pannelli DQM per creare, monitorare e controllare i canali dei gestori code remoti. Ogni gestore code ha un programma DQM per il controllo delle interconnessioni ai gestori code remoti compatibili.

Riferimenti correlati

[Configurazione di esempio - IBM MQ for IBM i](#)

[Esempio di pianificazione del canale di messaggi per IBM MQ for IBM i](#)

[Lavori di intercomunicazione su IBM i](#)

[Stati del canale su IBM i](#)

IBM i *Definizione di un collegamento TCP su IBM i*

È possibile definire una connessione TCP all'interno della definizione di canale utilizzando il campo Nome connessione.

La definizione del canale contiene un campo, CONNECTION NAME, che contiene l'indirizzo di rete TCP della destinazione o il nome host (ad esempio ABCHOST). L'indirizzo di rete TCP può essere in formato decimale puntato IPv4 (ad esempio, 127.0.0.1) o esadecimale IPv6 (ad esempio, 2001:DB8:0:0:0:0:0:0). Se CONNECTION NAME è un nome host o un server dei nomi, la tabella host IBM i viene utilizzata per convertire il nome host in un indirizzo host TCP.

Un numero di porta è richiesto per un indirizzo TCP completo; se questo numero non viene fornito, viene utilizzato il numero di porta predefinito 1414. All'estremità iniziale di una connessione (tipi di canale mittente, richiedente e server) è possibile fornire un numero di porta facoltativo per la connessione, ad esempio:

```
Connection name 127.0.0.1 (1555)
```

In questo caso, l'estremità di avvio tenta di collegarsi ad un programma di ricezione alla porta 1555.

MQ Adv. **CD** Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linuxo Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Utilizzo dell'opzione backlog del listener TCP

In TCP, le connessioni sono considerate incomplete a meno che non si verifichi un handshake a tre vie tra il server e il client. Queste connessioni vengono chiamate richieste di connessione in sospeso. Viene impostato un valore massimo per queste richieste di connessione in sospeso e può essere considerato un backlog di richieste in attesa sulla porta TCP affinché il listener accetti la richiesta.

Consultare “Utilizzo dell'opzione di backlog del listener TCP su IBM MQ for Multiplatforms” a pagina 279 per ulteriori informazioni e il valore specifico per IBM i.

Concetti correlati

[“Ricezione su TCP” a pagina 297](#)

I programmi del canale ricevente vengono avviati in risposta a una richiesta di avvio dal canale mittente. Per rispondere alla richiesta di avvio, è necessario avviare un programma listener per rilevare le richieste di rete in arrivo e avviare il canale associato. Questo programma listener viene avviato con il comando STRMQMLSR.

IBM i Ricezione su TCP

I programmi del canale ricevente vengono avviati in risposta a una richiesta di avvio dal canale mittente. Per rispondere alla richiesta di avvio, è necessario avviare un programma listener per rilevare le richieste di rete in arrivo e avviare il canale associato. Questo programma listener viene avviato con il comando STRMQMLSR.

È possibile avviare più di un listener per ciascun gestore code. Per impostazione predefinita, il comando STRMQMLSR utilizza la porta 1414, ma è possibile sovrascrivere questo valore. Per sovrascrivere l'impostazione predefinita, aggiungere le seguenti istruzioni al file qm.ini del gestore code selezionato. In questo esempio, il listener deve utilizzare la porta 2500:

```
TCP:
Port=2500
```

Il file qm.ini si trova in questa directory IFS: /QIBM/UserData/mqm/qmgrs/ *nome gestore code*.

Questo nuovo valore viene letto solo quando viene avviato il listener TCP. Se un listener è già in esecuzione, questa modifica non viene visualizzata da tale programma. Per utilizzare il nuovo valore, arrestare il listener ed emettere nuovamente il comando STRMQMLSR. Ora, ogni volta che si utilizza il comando STRMQMLSR, il listener assume il valore predefinito della nuova porta.

In alternativa, è possibile specificare un numero di porta diverso nel comando STRMQMLSR. Ad esempio:

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

Questa modifica rende il listener predefinito sulla nuova porta per la durata del lavoro listener.

Utilizzo dell'opzione TCP SO_KEEPALIVE

Se si desidera utilizzare l'opzione SO_KEEPALIVE (per ulteriori informazioni, consultare [“Verifica che l'altra estremità del canale sia ancora disponibile”](#) a pagina 242) è necessario aggiungere la seguente voce al file di configurazione del gestore code (qm.ini nella directory IFS, /QIBM/UserData/mqm/qmgrs/ *nome gestore code*):

```
TCP:
KeepAlive=yes
```

È quindi necessario immettere il seguente comando:

```
CFGTCP
```

Selezionare l'opzione 3 (Modifica attributi TCP). È ora possibile specificare un intervallo di tempo in minuti. È possibile specificare un valore compreso tra 1 e 40320 minuti; il valore predefinito è 120.

Utilizzo dell'opzione backlog del listener TCP

Quando si riceve su TCP, viene impostato un numero massimo di richieste di connessione in sospenso. Questo numero può essere considerato un *backlog* di richieste in attesa sulla porta TCP affinché il listener accetti la richiesta.

Il valore di backlog del listener predefinito su IBM i è 255. Se il backlog raggiunge questo valore, la connessione TCP viene rifiutata e non è possibile avviare il canale.

Per i canali MCA, ciò fa sì che il canale entri in uno stato RETRY e ritenti la connessione in seguito.

Per le connessioni client, il client riceve un codice motivo MQRC_Q_MGR_NOT_AVAILABLE da MQCONN e può ritentare la connessione in un secondo momento.

Tuttavia, per evitare questo errore, è possibile aggiungere una voce nel file qm.ini :

```
ListenerBacklog = n
```

Ciò sovrascrive il numero massimo predefinito di richieste in sospeso (255) per il listener TCP.

Nota: Alcuni sistemi operativi supportano un valore maggiore di quello predefinito. Se necessario, questo valore può essere utilizzato per evitare di raggiungere il limite di connessione.

IBM i **Definizione di una connessione LU 6.2 su IBM i**

Definire i dettagli delle comunicazioni LU 6.2 utilizzando un nome modalità, un nome TP e un nome connessione di una connessione LU 6.2 completa.

La fine iniziata del collegamento deve avere una definizione di voce di instradamento per completare questo oggetto CSI. Ulteriori informazioni sulla gestione delle richieste di lavoro dai sistemi LU 6.2 remoti sono disponibili in *IBM i Programming: Work Management Guide*.

Per informazioni, consultare il manuale *Multiplatform APPC Configuration Guide* e la seguente tabella.

Piattaforma remota	TPNAME
z/OS o MVS	Le stesse informazioni del lato corrispondente sul gestore code remoto.
IBM i	Uguale al valore di confronto nella specifica di instradamento sul sistema IBM i .
Sistemi AIX and Linux	Il programma di transazione richiamabile definito nella configurazione della LU remota 6.2 .
Windows	Come specificato nel comando Windows Esegui listener o nel programma di transazione richiamabile definito utilizzando TpSetup su Windows.

Se si dispone di più di un gestore code sullo stesso computer, verificare che i nomi TP nelle definizioni del canale siano univoci.

Concetti correlati

“Fine inizializzazione (mittente)” a pagina 298

Utilizzare il comando CRTMQMCHL per definire un canale di tipo trasporto *LU62.

“Fine avviata (Ricevente)” a pagina 301

Utilizzare il comando CRTMQMCHL per definire l'estremità di ricezione del collegamento del canale dei messaggi con tipo di trasporto *LU62.

IBM i **Fine inizializzazione (mittente)**

Utilizzare il comando CRTMQMCHL per definire un canale di tipo trasporto *LU62.

L'utilizzo dell'oggetto CSI è facoltativo in IBM MQ for IBM i V5.3 o versioni successive.

Il pannello di avvio finale viene mostrato nella Figura [LU 6.2 pannello di impostazione della comunicazione - avvio finale](#). Per ottenere il pannello completo come mostrato, premere F10 dal primo pannello.

```

Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . > WINSDOA1   Name
Library . . . . . > QSYS           Name, *CURLIB
Remote location . . . . . > WINSDOA1   Name
Transaction program . . . . . > MQSERIES

Text 'description' . . . . . *BLANK

Additional Parameters

Device . . . . . *LOC           Name, *LOC
Local location . . . . . *LOC           Name, *LOC, *NETATR
Mode . . . . . JSTMOD92        Name, *NETATR
Remote network identifier . . . *LOC           Name, *LOC, *NETATR, *NONE
Authority . . . . . *LIBCRTAUT   Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figura 34. Pannello di impostazione della comunicazione LU 6.2 - inizio

Completare i campi di inizio come segue:

Informazioni laterali

Assegnare a questa definizione un nome utilizzato per memorizzare l'oggetto di informazioni lato da creare, ad esempio WINSDOA1.

Nota: Per LU 6.2, il collegamento tra la definizione del canale di messaggi e la connessione di comunicazione è il campo **Nome connessione** della definizione del canale di messaggi all'estremità di invio. Questo campo contiene il nome dell'oggetto CSI.

Libreria

Il nome della libreria in cui è memorizzata questa definizione.

L'oggetto CSI deve essere disponibile in una libreria accessibile al programma che serve il canale dei messaggi, ad esempio QSYS, QMQM e QGPL.

Se il nome non è corretto, è mancante o non è possibile trovarlo, si verifica un errore all'avvio del canale.

Ubicazione remota

Specifica il nome dell'ubicazione remota con cui comunica il programma.

In breve, questo parametro obbligatorio contiene il nome dell'unità logica del partner sul sistema remoto, come definito nella descrizione dell'unità utilizzata per il collegamento di comunicazione tra i due sistemi.

Il nome dell' **Ubicazione remota** può essere trovato immettendo il comando DSPNETA sul sistema remoto e visualizzando il nome dell'ubicazione locale predefinito.

Programma di transazione

Specifica il nome (massimo 64 caratteri) del programma di transazione sul sistema remoto da avviare. Può essere un nome processo di transazione, un nome programma, il nome canale o una stringa di caratteri che corrisponde al **Valore di confronto** nella voce di instradamento.

Questo parametro è obbligatorio.

Nota: Per specificare i nomi dei programmi di transazioni di servizi SNA, immettere la rappresentazione esadecimale del nome del programma di transazioni di servizio. Ad esempio, per specificare un nome del programma di transazione servizio con una rappresentazione esadecimale di 21F0F0F1, immettere X'21F0F0F1'.

Ulteriori informazioni sui nomi dei programmi di transazione del servizio SNA sono disponibili nel manuale *SNA Transaction Programmer's Reference* per il tipo LU 6.2.

Se l'estremità di ricezione è un altro sistema IBM i , il nome del **Programma di transazione** viene utilizzato per far corrispondere l'oggetto CSI all'estremità di invio con la voce di instradamento all'estremità di ricezione. Questo nome deve essere univoco per ogni gestore code sul sistema IBM i di destinazione. Consultare il parametro **Program to call** in Initiated end (Receiver). Consultare anche il parametro **Dati di confronto: valore di confronto** nel pannello Aggiungi voce di instradamento.

Descrizione testo

Una descrizione (fino a 50 caratteri) per ricordare l'uso previsto di questa connessione.

Periferica

Specifica il nome della descrizione unità utilizzata per il sistema remoto. I valori possibili sono:

***LOC**

L'unità è determinata dal sistema.

Nome - periferica

Specificare il nome dell'unità associata all'ubicazione remota.

Ubicazione locale

Specifica il nome dell'ubicazione locale. I valori possibili sono:

***LOC**

Il nome dell'ubicazione locale è determinato dal sistema.

***NETATR**

Viene utilizzato il valore LCLLOCNAME specificato negli attributi di rete del sistema.

Nome - ubicazione - locale

Specificare il nome della propria ubicazione. Specificare l'ubicazione locale se si desidera indicare un nome di ubicazione specifica per l'ubicazione remota. Il nome dell'ubicazione può essere trovato utilizzando il comando DSPNETA.

Modalità

Specifica la modalità utilizzata per controllare la sessione. Questo nome è lo stesso di CPI (Common Programming Interface) - Communications Mode_Name. I valori possibili sono:

***NETATR**

Viene utilizzata la modalità negli attributi di rete.

BLANK

Vengono utilizzati otto caratteri vuoti.

Nome - modalità

Specificare un nome di modalità per l'ubicazione remota.

Nota: Poiché la modalità determina la priorità di trasmissione della sessione di comunicazioni, potrebbe essere utile definire diverse modalità a seconda della priorità dei messaggi inviati; ad esempio MQMODE_HI, MQMODE_MED e MQMODE_LOW. (È possibile avere più di un CSI che punta alla stessa ubicazione.)

Identificativo di rete remota

Specifica l'identificativo di rete remota utilizzato con l'ubicazione remota. I valori possibili sono:

***LOC**

Viene utilizzato l'ID di rete remota per l'ubicazione remota.

***NETATR**

Viene utilizzato l'identificativo di rete remota specificato negli attributi di rete.

***NONE**

La rete remota non ha nome.

ID - rete - remota

Specificare un ID rete remota. Utilizzare il comando DSPNETA nell'ubicazione remota per trovare il nome di questo ID di rete. È l'ID di rete locale nell'ubicazione remota.

Autorizzazione

Specifica l'autorizzazione che si sta fornendo agli utenti che non hanno un'autorizzazione specifica sull'oggetto, che non si trovano in un elenco di autorizzazioni e con un profilo di gruppo che non ha un'autorizzazione specifica sull'oggetto. I valori possibili sono:

***LIBCRTAUT**

L'autorizzazione pubblica per l'oggetto viene presa dal parametro CRTAUT della libreria specificata. Questo valore viene determinato al momento della creazione. Se il valore CRTAUT per la libreria cambia dopo la creazione dell'oggetto, il nuovo valore non influisce sugli oggetti esistenti.

***CHANGE**

L'autorizzazione alla modifica consente all'utente di eseguire funzioni di base sull'oggetto; tuttavia, l'utente non può modificare l'oggetto. L'autorizzazione alla modifica fornisce l'autorizzazione operativa sull'oggetto e su tutti i dati.

***ALL**

L'utente può eseguire tutte le operazioni tranne quelle limitate al proprietario o controllate dall'autorizzazione di gestione dell'elenco di autorizzazioni. L'utente può controllare l'esistenza dell'oggetto e specificare la sicurezza per l'oggetto, modificare l'oggetto ed eseguire funzioni di base sull'oggetto. L'utente può modificare la proprietà dell'oggetto.

***USE**

L'autorizzazione all'uso fornisce l'autorizzazione operativa sull'oggetto e l'autorizzazione alla lettura.

***EXCLUDE**

L'autorizzazione di esclusione impedisce all'utente di accedere all'oggetto.

Elenco autorizzazioni

Specificare il nome dell'elenco di autorizzazioni con l'autorizzazione utilizzata per le informazioni laterali.

IBM i *Fine avviata (Ricevente)*

Utilizzare il comando CRTMQMCHL per definire l'estremità di ricezione del collegamento del canale dei messaggi con tipo di trasporto *LU62.

Lasciare vuoto il campo CONNECTION NAME e verificare che i dettagli corrispondenti corrispondano all'estremità di invio del canale. Per i dettagli, vedi [Creazione di un canale](#).

Per consentire all'estremità di avvio di avviare il canale di ricezione, aggiungere una voce di instradamento ad un sottosistema alla fine avviata. Il sottosistema deve essere quello che assegna l'unità APPC utilizzata nelle sessioni LU 6.2 . Pertanto, deve avere una voce di comunicazioni valida per tale unità. La voce di instradamento richiama il programma che avvia l'estremità di ricezione del canale messaggi.

Utilizzare i comandi IBM i (ad esempio, ADDRTGE) per definire la fine del collegamento avviato da una sessione di comunicazioni.

Il pannello di fine inizializzato viene mostrato in [LU 6.2 pannello di impostazione della comunicazione - aggiungere la voce di instradamento](#).

```

Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description . . . . . QCMN      Name
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Routing entry sequence number . 1      1-9999
Comparison data:
Compare value . . . . . MQSERIES

Starting position . . . . . 37      1-80
Program to call . . . . . AMQCRC6B      Name, *RTGDTA
Library . . . . . QMAS400      Name, *LIBL, *CURLIB
Class . . . . . *SBSD      Name, *SBSD
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Maximum active routing steps . . *NOMAX      0-1000, *NOMAX
Storage pool identifier . . . . . 1      1-10

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

Figura 35. Pannello di impostazione delle comunicazioni LU 6.2 - fine avviata

Descrizione del sottosistema

Il nome del sottosistema in cui risiede questa definizione. Utilizzare il comando IBM i WRKSBSD per visualizzare e aggiornare la descrizione del sottosistema appropriata per la specifica di instradamento.

Numero di sequenza della voce di instradamento

Un numero univoco nel sottosistema per identificare questa definizione di comunicazione. È possibile utilizzare valori compresi tra 1 e 9999.

Dati di confronto: valore di confronto

Una stringa di testo da confrontare con la stringa ricevuta quando la sessione viene avviata da un parametro **Programma di transazione**, come mostrato nella [Figura 1](#). La stringa di caratteri deriva dal campo Programma di transazione del CSI mittente.

Dati di confronto: posizione iniziale

La posizione del carattere nella stringa in cui deve iniziare il confronto.

Nota: Il campo della posizione iniziale è la posizione del carattere nella stringa per il confronto e questa posizione è sempre 37.

Programma da chiamare

Il nome del programma che esegue il programma messaggi in entrata da chiamare per avviare la sessione.

Il programma, AMQCRC6A, viene richiamato per il gestore code predefinito. Questo programma viene fornito con IBM MQ for IBM i e imposta l'ambiente e richiama AMQCRS6A.

Per ulteriori gestori code:

- Ogni gestore code ha un programma richiamabile LU 6.2 specifico ubicato nella propria libreria. Questo programma viene denominato AMQCRC6B e viene generato automaticamente quando viene creato il gestore code.
- Ogni gestore code richiede una voce di instradamento specifica con dati di instradamento univoci da aggiungere. Questi dati di instradamento devono corrispondere al nome del **Programma di transazione** fornito dal sistema richiedente (consultare [Inizio fine \(Mittente\)](#)).

Un esempio viene mostrato in [LU 6.2 pannello di impostazione delle comunicazioni - visualizzare le voci di instradamento](#):

```

Display Routing Entries
System: MY400
Subsystem description: QCMN      Status: ACTIVE

Type options, press Enter.
5=Display details

Start
Opt  Seq Nbr  Program      Library      Compare Value  Pos
10   *RTGDTA           'QZSCSRVR'    37
20   *RTGDTA           'QZRCSRVR'    37
30   *RTGDTA           'QZHQTRG'    37
50   *RTGDTA           'QVPPRINT'    37
60   *RTGDTA           'QNPSRVR'     37
70   *RTGDTA           'QNMAPINGD'   37
80   QNMAREXECD  QSYS      'AREXECD'     37
90   AMQCR6A    QMQMBW    'MQSERIES'    37
100  *RTGDTA           'QTFDWNLD'   37
150  *RTGDTA           'QMFRCVR'    37

F3=Exit  F9=Display all detailed descriptions  F12=Cancel

```

Figura 36. Pannello di impostazione delle comunicazioni LU 6.2 - fine avviata

Nel pannello di configurazione delle comunicazioni LU 6.2 - voci di instradamento di visualizzazione, il numero di sequenza 90 rappresenta il gestore code predefinito e fornisce la compatibilità con le configurazioni delle release precedenti (ovvero, V3R2, V3R6, V3R7e V4R2) di IBM MQ for IBM i. Queste release consentono un solo gestore code. I numeri di sequenza 92 e 94 rappresentano due ulteriori gestori code denominati ALPHA e BETA creati con librerie QMALPHA e QMBETA.

Nota: È possibile avere più di una voce di instradamento per ciascun gestore code utilizzando dati di instradamento differenti. Queste voci forniscono l'opzione di priorità di lavoro differenti a seconda delle classi utilizzate.

Classe

Il nome e la libreria della classe utilizzata per i passi avviati tramite questa specifica di instradamento. La classe definisce gli attributi dell'ambiente di esecuzione del passo di instradamento e specifica la priorità lavoro. È necessario specificare una voce di classe appropriata. Utilizzare, ad esempio, il comando WRKCLS per visualizzare le classi esistenti o per creare una classe. Ulteriori informazioni sulla gestione delle richieste di lavoro dai sistemi LU 6.2 remoti sono disponibili in *IBM i Programming: Work Management Guide*.

Nota sulla gestione del lavoro

Il lavoro AMQCR6A non è in grado di sfruttare le normali funzioni di gestione lavoro di IBM i documentate in Gestione lavoro poiché non viene avviato nello stesso modo di altri lavori IBM MQ. Per modificare le proprietà di runtime dei lavori del destinatario LU62, è possibile apportare una delle seguenti modifiche:

- Modificare la descrizione della classe specificata nella specifica di instradamento per il lavoro AMQCR6A
- Modificare la descrizione del lavoro sulla specifica di comunicazioni

Consultare *IBM i Programming: Work Management Guide* per ulteriori informazioni sulla configurazione dei lavori di comunicazione.

Configurazione di un cluster di gestore code

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. È possibile definire componenti cluster e creare e gestire cluster.

Prima di iniziare

Per un'introduzione ai concetti di cluster, consultare [Cluster](#).

Quando si progetta il cluster del gestore code, è necessario prendere alcune decisioni. Vedi [Cluster di esempio](#) e [Progettazione dei cluster](#).

Attività correlate

[“Spostamento di una definizione di argomento del raggruppamento in un gestore code differente” a pagina 456](#)

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Riferimenti correlati

[Elimina argomento](#)

Definizione dei componenti di un cluster

I cluster sono composti da gestori code, canali cluster e code cluster. È possibile definire le code del cluster e modificare alcuni aspetti degli oggetti cluster predefiniti. È possibile ottenere informazioni di configurazione e stato sui canali definiti automaticamente e sulla relazione tra i singoli canali mittente del cluster e le code di trasmissione.

Consultare i seguenti argomenti secondari per informazioni sulla definizione di ciascuno dei componenti cluster:

Concetti correlati

[Componenti di un cluster](#)

[Canali cluster](#)

Attività correlate

[Definizione degli argomenti del cluster](#)

[“Configurazione di un nuovo cluster” a pagina 318](#)

Seguire queste istruzioni per configurare il cluster di esempio. Istruzioni separate descrivono l'impostazione del cluster su TCP/IP, LU 6.2e con una o più code di trasmissione. Verificare il funzionamento del cluster inviando un messaggio da un gestore code all'altro.

[“Aggiunta di un gestore code a un cluster” a pagina 329](#)

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code e gli argomenti del cluster vengono trasferiti utilizzando la singola coda di trasmissione del cluster SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Definizione di code cluster

Una coda cluster è una coda ospitata da un gestore code cluster e resa disponibile ad altri gestori code del cluster. Definire una coda cluster come coda locale sul gestore code cluster in cui si trova la coda. Specificare il nome del cluster a cui appartiene la coda.

Il seguente esempio mostra un comando **runmqsc** per definire una coda cluster con l'opzione CLUSTER :

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

Una definizione di coda cluster viene pubblicizzata in altri gestori code nel cluster. Gli altri gestori code nel cluster possono inserire i messaggi in una coda cluster senza la necessità di una definizione di coda remota corrispondente. Una coda cluster può essere pubblicizzata in più di un cluster utilizzando un elenco dei nomi di cluster.

Quando una coda viene pubblicizzata, qualsiasi gestore code del cluster può inserire dei messaggi al suo interno. Per inserire un messaggio, il gestore code deve scoprire, dai repository completi, la posizione in cui è ospitata la coda. Aggiunge quindi alcune informazioni di instradamento al messaggio e inserisce tale messaggio su una coda di trasmissione del cluster.

Una coda del cluster può essere una coda condivisa dai membri di un gruppo di condivisione di code in IBM MQ for z/OS.

Associazione

È possibile creare un cluster in cui più di un gestore code ospita un'istanza della stessa coda cluster. Assicurarsi che tutti i messaggi in sequenza vengano inviati alla stessa istanza della coda. È possibile associare una serie di messaggi a una particolare coda utilizzando l'opzione `MQOO_BIND_ON_OPEN` sulla chiamata `MQOPEN`.

Code di trasmissione cluster

Un gestore code può memorizzare i messaggi per altri gestori code di un cluster su più code di trasmissione. È possibile configurare un gestore code per memorizzare messaggi su più code di trasmissione cluster in due diversi modi. Se si imposta l'attributo del gestore code **DEFCLXQ** su `CHANNEL`, viene creata automaticamente una coda di trasmissione cluster differente da `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` per ogni canale mittente del cluster. Se si imposta l'opzione della coda di trasmissione `CLCHNAME` per trovare la corrispondenza con uno o più canali mittenti del cluster, il gestore code può memorizzare i messaggi per i canali corrispondenti su tale coda di trasmissione.



Attenzione: Se si stanno utilizzando delle `SYSTEM.CLUSTER.TRANSMIT.QUEUES` dedicate con un gestore code che era stato aggiornato da una versione del prodotto antecedente a IBM WebSphere MQ 7.5, assicurarsi che la `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` abbia l'opzione SHARE/NOSHARE impostata su **SHARE**.

Un messaggio per una coda cluster su un gestore code differente viene posizionato su una coda di trasmissione cluster prima di essere inviato. Un canale mittente del cluster trasferisce i messaggi da una coda di trasmissione del cluster ai canali riceventi del cluster su altri gestori code. Per impostazione predefinita, una coda di trasmissione cluster definita dal sistema contiene tutti i messaggi che devono essere trasferiti ad altri gestori code cluster. La coda è denominata `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Un gestore code che fa parte di un cluster può inviare messaggi su questa coda di trasmissione del cluster a qualsiasi altro gestore code nello stesso cluster.

Una definizione per la singola coda `SYSTEM.CLUSTER.TRANSMIT.QUEUE` viene creata per impostazione predefinita su ogni gestore code Multiplatforms. Su z/OS, la definizione può essere definita con l'esempio fornito **CSQ4INSX**.

È possibile configurare un gestore code per trasferire i messaggi ad altri gestori code con cluster utilizzando più code di trasmissione. È possibile definire manualmente ulteriori code di trasmissione del cluster oppure fare in modo che il gestore code crei automaticamente le code.

Per creare automaticamente le code dal gestore code, modificare l'attributo del gestore code `DEFCLXQ` da `SCTQ` a `CHANNEL`. Il risultato è che il gestore code crea una singola coda di trasmissione cluster per ogni canale mittente del cluster creato. Le code di trasmissione vengono create come code dinamiche permanenti dalla coda modello, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Il nome di ogni coda dinamica permanente è `SYSTEM.CLUSTER.TRANSMIT.ChannelName`. Il nome del canale mittente del cluster a cui è associato ogni coda di trasmissione del cluster dinamico permanente è impostato nell'attributo della coda di trasmissione locale `CLCHNAME`. I messaggi per i gestori code con cluster remoti vengono posizionati sulla coda di trasmissione cluster dinamica permanente per il canale mittente del cluster associato, piuttosto che su `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Per creare le code di trasmissione del cluster manualmente, creare una coda locale con l'attributo `USAGE` impostato su `XMITQe` e l'attributo `CLCHNAME` impostato su un nome di canale generico che si risolve in uno o più canali mittente del cluster; consultare [ClusterChannelName](#). Se si creano manualmente le code di trasmissione del cluster, è possibile associare la coda di trasmissione a un singolo canale mittente del cluster o a più canali mittente del cluster. L'attributo `CLCHNAME` è un nome generico, che significa che è possibile inserire più caratteri jolly, "*", nel nome.

Fatta eccezione per i canali mittenti del cluster iniziali creati manualmente per collegare un gestore code a un repository completo, i canali mittenti del cluster vengono creati automaticamente. Vengono creati

automaticamente quando è presente un messaggio da trasferire a un gestore code del cluster. Vengono creati con lo stesso nome del canale ricevente del cluster che riceve i messaggi cluster per quel cluster particolare sul gestore code di destinazione.

Se si segue una convenzione di denominazione per i canali riceventi del cluster, è possibile definire un valore generico per CLCHNAME che filtra i diversi tipi di messaggi del cluster in code di trasmissione differenti. Ad esempio, se si segue la convenzione di denominazione per i canali riceventi del cluster di *ClusterName*. *QmgrName*, il nome generico *ClusterName*. * filtra i messaggi per i diversi cluster su code di trasmissione differenti. È necessario definire le code di trasmissione manualmente e impostare CLCHNAME in ciascuna coda di trasmissione su *ClusterName*. *.

Le modifiche all'associazione delle code di trasmissione del cluster ai canali mittenti del cluster non hanno effetto immediato. La coda di trasmissione attualmente associata che un canale mittente del cluster sta gestendo potrebbe contenere messaggi in fase di trasferimento dal canale mittente del cluster. Solo quando nessun messaggio sulla coda di trasmissione attualmente associata viene elaborato da un canale mittente del cluster, il gestore code può modificare l'associazione del canale mittente del cluster di una coda di trasmissione diversa. Ciò può verificarsi quando nessun messaggio rimane sulla coda di trasmissione per essere elaborato dal canale mittente del cluster o quando l'elaborazione dei messaggi è sospesa e il canale mittente del cluster non ha alcun messaggio "incompleto". Quando ciò si verifica, tutti i messaggi non elaborati per il canale mittente del cluster vengono trasferiti alla coda di trasmissione appena associata e l'associazione del canale mittente del cluster viene modificata.

È possibile creare una definizione di coda remota che si risolva in una coda di trasmissione cluster. Nella definizione, il gestore code QMX si trova nello stesso cluster del gestore code locale e non esiste alcuna coda di trasmissione, QMX.

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

Durante la risoluzione del nome della coda, la coda di trasmissione del cluster ha la precedenza sulla coda di trasmissione predefinita. Un messaggio inserito in A viene memorizzato sulla coda di trasmissione cluster e quindi inviato alla coda remota B su QMX.

I gestori code possono anche comunicare con altri gestori code che non fanno parte del cluster. È necessario definire i canali e una coda di trasmissione per l'altro gestore code, come in un ambiente di accodamento distribuito.

Nota: Le applicazioni devono scrivere nelle code che si risolvono nella coda di trasmissione cluster e non devono scrivere direttamente nella coda di trasmissione cluster.

Definizione automatica delle code remote

Un gestore code in un cluster non necessita di una definizione di coda remota per le code remote nel cluster. Il gestore code del cluster trova l'ubicazione di una coda remota dal repository completo. Aggiunge le informazioni di instradamento al messaggio e le inserisce nella coda di trasmissione cluster. IBM MQ crea automaticamente una definizione equivalente ad una definizione di coda remota in modo che il messaggio possa essere inviato.

Non è possibile modificare o eliminare una definizione di coda remota creata automaticamente. Tuttavia, utilizzando il comando `DISPLAY QUEUE runmqsc` con l'attributo `CLUSINFO`, è possibile visualizzare tutte le code locali su un gestore code e tutte le code cluster, incluse le code cluster su gestori code remoti. Ad esempio:

```
DISPLAY QUEUE(*) CLUSINFO
```

Concetti correlati

[Code cluster](#)

[Come scegliere quale tipo di coda di trasmissione del cluster utilizzare](#)

Riferimenti correlati

[Nome ClusterChannel\(MQCHAR20\)](#)

Utilizzo dei canali mittenti del cluster definiti in modo automatico

Dopo aver introdotto un gestore code in un cluster effettuando le definizioni CLUSSDR e CLUSRCVR iniziali, IBM MQ crea automaticamente altre definizioni del canale mittente del cluster quando richiesto per spostare i messaggi in un altro gestore code del cluster. È possibile visualizzare le informazioni sui canali mittenti del cluster definiti automaticamente, ma non è possibile modificarle. Per modificarne il funzionamento, è possibile utilizzare un'uscita di definizione automatica del canale.

Prima di iniziare

Per un'introduzione ai canali definiti automaticamente, consultare [Canali mittenti del cluster definiti automaticamente](#).

Informazioni su questa attività

I canali mittente del cluster definiti automaticamente vengono creati dal cluster come e quando necessario e rimangono attivi fino a quando non vengono arrestati utilizzando le normali regole di intervallo di disconnessione.

I canali mittenti del cluster (CLUSSDR) possono essere definiti automaticamente per spostare i messaggi dell'applicazione e i messaggi di gestione del cluster interni. Ad esempio, in un cluster di [pubblicazione / sottoscrizione](#) (uno in cui è stato definito un argomento del cluster), i canali possono essere definiti tra repository parziali per consentire lo scambio dello stato di 'sottoscrizione proxy'. Quando non sono richiesti (inattivi) per un periodo di tempo prolungato, i CLUSSDR definiti automaticamente vengono rimossi dalla cache di un repository parziale delle informazioni sul cluster e non sono più visibili su tale gestore code.

Multi Su Multipiattaforme, OAM (object authority manager) non è consapevole dell'esistenza di canali mittenti del cluster definiti automaticamente. Se si immettono i comandi **start**, **stop**, **ping**, **reseto** **resolve** su un canale mittente del cluster definito automaticamente, OAM controlla se si è autorizzati a eseguire la stessa azione sul canale ricevente del cluster corrispondente.

z/OS Su z/OS, è possibile proteggere un canale mittente del cluster definito automaticamente come qualsiasi altro canale.

Procedura

- Visualizzare le informazioni sui canali definiti automaticamente per un determinato gestore code cluster.

Non è possibile visualizzare automaticamente i canali definiti utilizzando il comando DISPLAY CHANNEL **runmqsc**. Per visualizzare i canali definiti automaticamente utilizzare il seguente comando:

```
DISPLAY CLUSQMGR(qMgrName)
```

- Visualizza lo stato del canale definito automaticamente per un determinato CLUSRCVR.

Per visualizzare lo stato del canale CLUSSDR definito automaticamente corrispondente alla definizione di canale CLUSRCVR creata, utilizzare il seguente comando:

```
DISPLAY CHSTATUS(channelname)
```

- Utilizzare un'uscita di definizione automatica del canale per modificare il comportamento di un canale definito automaticamente.

È possibile utilizzare l'uscita di definizione automatica del canale IBM MQ se si desidera scrivere un programma di uscita utente per personalizzare un canale mittente del cluster o un canale ricevente del cluster. Ad esempio, è possibile utilizzare l'uscita di definizione automatica del canale in un ambiente cluster per apportare le seguenti modifiche:

- Personalizzare le definizioni delle comunicazioni, ovvero i nomi SNA LU6.2.

- Aggiungere o rimuovere altre uscite, ad esempio uscite di sicurezza.
- Modificare i nomi delle uscite canale.

Il nome dell'uscita del canale CLUSSDR viene generato automaticamente dalla definizione di canale CLUSRCVR e, pertanto, potrebbe non essere appropriato per le proprie esigenze, soprattutto se le due estremità del canale si trovano su piattaforme differenti.

Il formato dei nomi di uscita è diverso su piattaforme differenti. Ad esempio:

- **z/OS** Sulla piattaforma z/OS , il formato del parametro SCYEXIT (*nome uscita di sicurezza*) è SCYEXIT(' SECEXIT ')
- **Windows** Su piattaforme Windows , il formato del parametro SCYEXIT (*nome uscita di sicurezza*) è SCYEXIT(' *drive:\path\library (secexit)* ')

Nota: **z/OS** Se non è presente alcuna uscita di definizione automatica del canale, il gestore code z/OS ricava il nome dell'uscita del canale CLUSSDR dalla definizione del canale CLUSRCVR sull'altra estremità del canale. Per derivare il nome dell'uscita z/OS da un nome nonz/OS , viene utilizzato l'algoritmo seguente:

- I nomi di uscita su Multipiattaforme sono nel formato generale *percorso/libreria (funzione)*.
- Se è presente *function* , vengono utilizzati fino a otto caratteri.
- Altrimenti, vengono utilizzati fino a otto caratteri della *libreria* .

Ad esempio:

- /var/mqm/exits/myExit.so(MsgExit) converte in MSGEXIT
- /var/mqm/exits/myExit converte in MYEXIT
- /var/mqm/exits/myExit.so(ExitLongName) converte in EXITLONG

- Se il cluster deve utilizzare **PROPCTL** per rimuovere le intestazioni dell'applicazione come RFH2 dai messaggi che vanno da un gestore code IBM MQ a un gestore code su una versione precedente del prodotto, è necessario scrivere un'uscita di definizione automatica del canale che imposta **PROPCTL** su un valore di NONE.
- Utilizzare l'attributo del canale LOCLADDR per controllare gli aspetti dell'indirizzamento.
 - Per abilitare un canale in uscita (TCP) per utilizzare un particolare indirizzo IP, porta o intervallo di porte, utilizzare l'attributo del canale LOCLADDR. Ciò è utile se si dispone di più di una scheda di rete e si desidera che un canale ne utilizzi una specifica per le comunicazioni in uscita.
 - Per specificare un indirizzo IP virtuale sui canali CLUSSDR , utilizzare l'indirizzo IP da LOCLADDR su un CLUSSDRdefinito manualmente. Per specificare l'intervallo di porte, utilizzare l'intervallo di porte da CLUSRCVR.
 - Se un cluster deve utilizzare LOCLADDR per ottenere i canali di comunicazione in uscita da collegare a un indirizzo IP specifico, è possibile scrivere un'uscita di definizione automatica del canale per forzare il valore LOCLADDR in uno qualsiasi dei relativi canali CLUSSDR definiti automaticamente. È necessario specificarlo anche nel canale definito manualmente CLUSSDR .
 - Inserire un numero di porta o un intervallo di porte in LOCLADDR di un canale CLUSRCVR , se si desidera che tutti i gestori code in un cluster utilizzino una porta o un intervallo di porte specifici per tutte le relative comunicazioni in uscita.

Nota: non inserire un indirizzo IP nel campo LOCLADDR di un canale CLUSRCVR , a meno che tutti i gestori code non si trovino sullo stesso server. L'indirizzo IP LOCLADDR viene propagato ai canali CLUSSDR definiti automaticamente di tutti i gestori code che si connettono utilizzando il canale CLUSRCVR .

Multi Su Multipiattaforme, è possibile impostare un valore di indirizzo locale predefinito che viene utilizzato per tutti i canali mittenti che non hanno un indirizzo locale definito. Il valore predefinito viene definito impostando la variabile di ambiente MQ_LCLADDR prima di avviare il gestore code. Il formato del valore corrisponde a quello dell'attributo MQSC LOCLADDR.

Riferimenti correlati

[Indirizzo locale \(LOCLADDR\)](#)

Utilizzo degli oggetti cluster predefiniti

È possibile modificare le definizioni di canale predefinite come qualsiasi altra definizione di canale, eseguendo i comandi MQSC o PCF. Non modificare le definizioni di coda predefinite, tranne per SYSTEM.CLUSTER.HISTORY.QUEUE.

Per un elenco completo di questi oggetti, consultare [Oggetti cluster predefiniti](#). Il seguente elenco include solo gli oggetti che è possibile modificare.


SYSTEM.CLUSTER.HISTORY.QUEUE

Ogni gestore code in un cluster ha una coda locale denominata SYSTEM.CLUSTER.HISTORY.QUEUE. SYSTEM.CLUSTER.HISTORY.QUEUE viene utilizzato per memorizzare la cronologia delle informazioni sullo stato del cluster per scopi di servizio.

Nelle impostazioni dell'oggetto predefinito, SYSTEM.CLUSTER.HISTORY.QUEUE è impostata su PUT (ENABLED). Per eliminare la raccolta cronologica, modificare l'impostazione in PUT (DISABLED).

SYSTEM.CLUSTER.TRANSMIT.QUEUE

Ogni Gestore code ha una definizione per una coda locale denominata SYSTEM.CLUSTER.TRANSMIT.QUEUE. SYSTEM.CLUSTER.TRANSMIT.QUEUE è la coda di trasmissione predefinita per tutti i messaggi a tutte le code e i gestori code all'interno dei cluster. È possibile modificare la coda di trasmissione predefinita per ogni canale mittente del cluster in SYSTEM.CLUSTER.TRANSMIT.ChannelName, modificando l'attributo del gestore code DEFXMITQ

, tranne in z/OS. Non è possibile eliminare SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Viene utilizzato anche per definire i controlli di autorizzazione se la coda di trasmissione predefinita utilizzata è SYSTEM.CLUSTER.TRANSMIT.QUEUE o SYSTEM.CLUSTER.TRANSMIT.ChannelName.

Concetti correlati

[Oggetti cluster predefiniti](#)

Utilizzo delle code di trasmissione del cluster e dei canali mittente del cluster

I messaggi tra i gestori code con cluster vengono memorizzati nelle code di trasmissione cluster e inoltrati dai canali mittenti del cluster. In qualsiasi momento, un canale mittente del cluster è associato a una coda di trasmissione. Se si modifica la configurazione del canale, questa potrebbe passare a una coda di trasmissione diversa al successivo avvio. L'elaborazione di questo switch è automatizzata e transazionale.

Eseguire questo comando MQSC per visualizzare le code di trasmissione a cui sono associati i canali mittenti del cluster:

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.  
CHANNEL(TO.QM2)          CHLTYPE(CLUSSDR)  
CONNAME(9.146.163.190(1416))  CURRENT  
RQMNAME(QM2)             STATUS(STOPPED)  
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

La coda di trasmissione visualizzata nello stato del canale salvato di un canale mittente del cluster arrestato potrebbe cambiare quando il canale viene riavviato. La [“Selezione di code di trasmissione predefinite per canali mittente del cluster”](#) a pagina 310 descrive il processo di selezione di una coda di trasmissione predefinita; [“Selezione di code di trasmissione definite manualmente dai canali mittente del cluster”](#) a pagina 311 descrive il processo di selezione di una coda di trasmissione definita manualmente.

Quando un canale mittente del cluster viene avviato, ricontrolla la sua associazione con le code di trasmissione. Se la configurazione delle code di trasmissione o i valori predefiniti del gestore code vengono modificati, è possibile che il canale venga riassociato a una coda di trasmissione differente. Se il canale viene riavviato con una coda di trasmissione differente come risultato di una modifica della configurazione, viene eseguito un processo di trasferimento dei messaggi alla coda di trasmissione

appena associata. [“Come funziona il processo di commutazione del canale mittente del cluster in una coda di trasmissione differente”](#) a pagina 312 descrive il processo di trasferimento di un canale mittente del cluster da una coda di trasmissione ad un altro.

Il comportamento dei canali mittente del cluster è diverso da quello dei canali mittente e server. Rimangono associate alla stessa coda di trasmissione fino a quando l'attributo del canale **XMITQ** non viene modificato. Se si modifica l'attributo della coda di trasmissione su un canale mittente o server e lo si riavvia, i messaggi non vengono trasferiti dalla vecchia coda di trasmissione a quella nuova.

Un'altra differenza tra i canali mittente del cluster e i canali mittente o server è che più canali mittente del cluster possono aprire una coda di trasmissione del cluster, ma solo un canale mittente o server può aprire una coda di trasmissione normale. È possibile che i canali mittente del cluster non condividano le code di trasmissione. L'esclusività non viene applicata; è un risultato della configurazione. È possibile configurare il percorso di un messaggio in un cluster in modo che non condivida alcuna coda di trasmissione o canale con i messaggi che fluiscono tra altre applicazioni. Consultare [Cluster: Pianificazione della configurazione delle code di trasmissione cluster](#) e [“Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway”](#) a pagina 362.

Selezione di code di trasmissione predefinite per canali mittente del cluster

Una coda di trasmissione del cluster è una coda predefinita del sistema, con un nome che inizia con SYSTEM . CLUSTER . TRANSMIT, oppure una coda definita manualmente. Un canale mittente del cluster viene associato a una coda di trasmissione del cluster in uno dei due modi: dal meccanismo della coda di trasmissione del cluster predefinito o dalla configurazione manuale.

La coda di trasmissione del cluster predefinita è impostata come attributo del gestore code, **DEFCLXQ**. Il valore è SCTQ o CHANNEL. I gestori code nuovi e migrati sono impostati su SCTQ. È possibile modificare il valore in CHANNEL.

Se SCTQ è impostato, la coda di trasmissione del cluster predefinita è SYSTEM . CLUSTER . TRANSMIT . QUEUE. Ogni canale mittente del cluster può aprire questa coda. I canali mittenti del cluster che aprono la coda sono quelli che non sono associati alle code di trasmissione del cluster definite manualmente.

Se CHANNEL è impostato, il gestore code può creare una coda di trasmissione dinamica permanente separata per ogni canale mittente del cluster. Ogni coda è denominata SYSTEM . CLUSTER . TRANSMIT . *ChannelName* e viene creata dalla coda modello, SYSTEM . CLUSTER . TRANSMIT . MODEL . QUEUE. Ogni canale mittente del cluster non associato a una code di trasmissione cluster definita manualmente è associato a una coda di trasmissione cluster dinamica permanente. La coda viene creata dal gestore code quando richiede una coda di trasmissione cluster separata per la destinazione cluster servita da questo canale mittente del cluster e non esiste alcuna coda.

Alcune destinazioni cluster possono essere servite da canali mittenti del cluster associati a code di trasmissione definite manualmente e altre dalla coda o dalle code predefinite. Nell'associazione dei canali mittenti del raggruppamento con le code di trasmissione, le code di trasmissione definite manualmente hanno sempre la precedenza sulle code di trasmissione predefinite.

La precedenza delle code di trasmissione cluster è illustrata in [Figura 37 a pagina 311](#). L'unico canale mittente del cluster non associato ad una coda di trasmissione cluster definita manualmente è CS . QM1. Non è associato a una coda di trasmissione definita manualmente, perché nessuno dei nomi di canale nell'attributo **CLCHNAME** delle code di trasmissione corrisponde a CS . QM1.

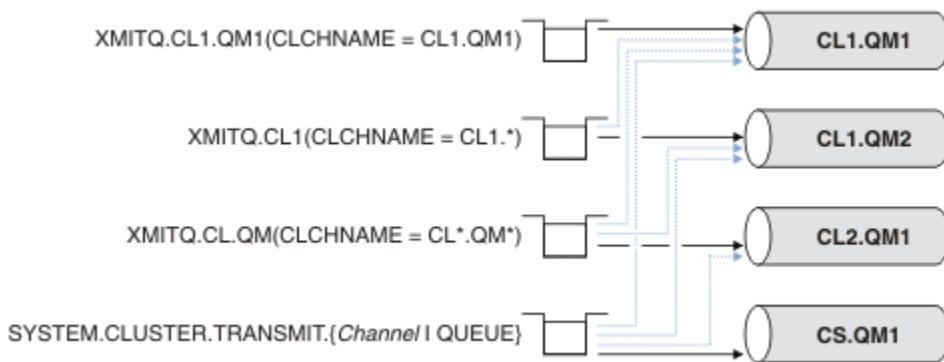


Figura 37. Precedenza coda di trasmissione / canale mittente del cluster

Selezione di code di trasmissione definite manualmente dai canali mittente del cluster

Una coda definita manualmente ha l'attributo della coda di trasmissione **USAGE** impostato su XMITQ e l'attributo del nome del canale cluster **CLCHNAME** impostato su un nome canale specifico o generico.

Se il nome nell'attributo della coda **CLCHNAME** corrisponde a un nome canale mittente del cluster, il canale è associato alla coda. Il nome è una corrispondenza esatta, se il nome non contiene caratteri jolly, oppure la corrispondenza migliore, se il nome contiene caratteri jolly.

Se le definizioni **CLCHNAME** su più code di trasmissione corrispondono allo stesso canale mittente del cluster, si dice che le definizioni si sovrappongono. Per risolvere l'ambiguità, esiste un ordine di precedenza tra le corrispondenze. Le corrispondenze esatte hanno sempre la precedenza. [Figura 37 a pagina 311](#) mostra le associazioni tra le code di trasmissione e i canali mittente del cluster. Le frecce nere mostrano le associazioni effettive e le frecce grigie, le associazioni potenziali. L'ordine di precedenza delle code di trasmissione in [Figura 37 a pagina 311](#) è:

XMITQ.CL1.QM1

La coda di trasmissione XMITQ.CL1.QM1 ha il proprio attributo **CLCHNAME** impostato su CL1.QM1. La definizione dell'attributo **CLCHNAME**, CL1.QM1, non ha caratteri jolly e ha la precedenza su qualsiasi altro attributo CLCHNAME, definito su altre code di trasmissione, che corrispondono ai caratteri jolly. Il gestore code memorizza qualsiasi messaggio cluster che deve essere trasferito dal canale mittente del cluster CL1.QM1 sulla coda di trasmissione XMITQ.CL1.QM1. L'unica eccezione è se per più code di trasmissione il relativo attributo **CLCHNAME** è impostato su CL1.QM1. In tal caso, il gestore code memorizza i messaggi per il canale mittente del cluster CL1.QM1 su una qualsiasi di tali code. Seleziona una coda in modo arbitrario all'avvio del canale. Potrebbe selezionare una coda differente quando il canale viene riavviato.

XMITQ.CL1

La coda di trasmissione XMITQ.CL1 ha il proprio attributo **CLCHNAME** impostato su CL1.*. La definizione dell'attributo **CLCHNAME**, CL1.*, ha un carattere jolly finale, che corrisponde al nome di qualsiasi canale mittente del cluster che inizia con CL1.. Il gestore code memorizza qualsiasi messaggio cluster che deve essere trasferito da qualsiasi canale mittente del cluster il cui nome inizia con CL1. nella coda di trasmissione XMITQ.CL1, a meno che non vi sia una coda di trasmissione con una corrispondenza più specifica, come la coda XMITQ.CL1.QM1. Un carattere jolly finale rende la definizione meno specifica di una definizione senza caratteri jolly e più specifica di una definizione con più caratteri jolly o caratteri jolly seguiti da più caratteri finali.

XMITQ.CL.QM

XMITQ.CL.QM è il nome della coda di trasmissione con il suo attributo **CLCHNAME** impostato su CL*.QM*. La definizione di CL*.QM* ha due caratteri jolly, che corrispondono al nome di qualsiasi canale mittente del cluster che inizia con CL. e che include o termina con QM. La corrispondenza è meno specifica di una corrispondenza con un carattere jolly.

SYSTEM.CLUSTER.TRANSMIT. *channelName* | QUEUE

Se nessuna coda di trasmissione ha un attributo **CLCHNAME** che corrisponde al nome del canale mittente del cluster che il gestore code deve utilizzare, il gestore code utilizza la coda di trasmissione del cluster predefinita. La coda di trasmissione del cluster predefinita è la coda di trasmissione del cluster del sistema singolo, SYSTEM.CLUSTER.TRANSMIT.QUEUE, o una coda di trasmissione del cluster del sistema creata dal gestore code per uno specifico canale mittente del cluster, SYSTEM.CLUSTER.TRANSMIT. *channelName*. La coda predefinita dipende dall'impostazione del gestore code **DEFXMITQ**.

Suggerimento: A meno che non si abbia una chiara necessità di sovrapposizioni di definizioni, evitarle in quanto possono portare a configurazioni complicate e difficili da comprendere.

Come funziona il processo di commutazione del canale mittente del cluster in una coda di trasmissione differente

Per modificare l'associazione dei canali mittenti del cluster con le code di trasmissione del cluster, modificare il parametro **CLCHNAME** di qualsiasi coda di trasmissione o il parametro del gestore code **DEFCLXQ** in qualsiasi momento. Nulla accade immediatamente. Le modifiche si verificano solo quando viene avviato un canale. Quando viene avviato, verifica se continuare ad inoltrare i messaggi dalla stessa coda di trasmissione. Tre tipi di modifica modificano l'associazione di un canale mittente del cluster con una coda di trasmissione.

1. Ridefinire il parametro **CLCHNAME** della coda di trasmissione a cui è attualmente associato il canale mittente del cluster in modo che sia meno specifico o vuoto oppure eliminare la coda di trasmissione del cluster quando il canale viene arrestato.

Alcune altre code di trasmissione del cluster potrebbero ora essere una migliore corrispondenza per il nome del canale. Oppure, se nessun'altra coda di trasmissione corrisponde al nome del canale mittente del cluster, l'associazione deve tornare alla coda di trasmissione predefinita.

2. Ridefinizione del parametro **CLCHNAME** di qualsiasi altra coda di trasmissione cluster o aggiunta di una coda di trasmissione cluster.

Il parametro **CLCHNAME** di un'altra coda di trasmissione potrebbe ora essere una corrispondenza migliore per il canale mittente del cluster rispetto alla coda di trasmissione a cui è attualmente associato il canale mittente del cluster. Se il canale mittente del cluster è attualmente associato a una coda di trasmissione cluster predefinita, potrebbe essere associato a una coda di trasmissione cluster definita manualmente.

3. Se il canale mittente del cluster è attualmente associato a una coda di trasmissione del cluster predefinita, modificare il parametro del gestore code **DEFCLXQ**.

Se l'associazione di un canale mittente del cluster cambia, quando il canale viene avviato passa la sua associazione alla nuova coda di trasmissione. Durante lo switch, garantisce che nessun messaggio venga perso. I messaggi vengono trasferiti alla nuova coda di trasmissione nell'ordine in cui il canale trasferisce i messaggi al gestore code remoto.

Attenzione: In comune con qualsiasi inoltro di messaggi in un cluster, è necessario inserire i messaggi in gruppi per garantire che i messaggi che devono essere consegnati in ordine vengano consegnati in ordine. In rare occasioni, i messaggi possono essere fuori ordine in un cluster.

Il processo di commutazione passa attraverso i seguenti passi transazionali. Se il processo di commutazione viene interrotto, la fase transazionale corrente viene ripresa quando il canale viene riavviato.

Passo 1 - Elaborazione dei messaggi dalla coda di trasmissione originale

Il canale mittente del cluster è associato alla nuova coda di trasmissione, che potrebbe condividere con altri canali mittente del cluster. I messaggi per il canale mittente del cluster continuano ad essere inseriti nella coda di trasmissione originale. Un processo di commutazione di transizione trasferisce i messaggi dalla coda di trasmissione originale alla nuova coda di trasmissione. Il canale mittente del cluster inoltra i messaggi dalla nuova coda di trasmissione al canale ricevente del cluster. Lo stato del canale mostra il canale mittente del cluster ancora associato alla vecchia coda di trasmissione.

Il processo di commutazione continua a trasferire anche i messaggi appena arrivati. Questo passo continua fino a quando il numero di messaggi rimanenti che devono essere inoltrati dal processo di commutazione non raggiunge lo zero. Quando il numero di messaggi raggiunge lo zero, la procedura passa al punto 2.

Durante il passo 1, l'attività del disco per il canale aumenta. I messaggi persistenti vengono sottoposti a commit dalla prima coda di trasmissione e sulla seconda coda di trasmissione. Questa attività disco è in aggiunta ai messaggi di cui viene eseguito il commit quando vengono inseriti e rimossi dalla coda di trasmissione come parte del trasferimento dei messaggi normalmente. Idealmente, nessun messaggio arriva durante il processo di commutazione, in modo che la transizione possa avvenire il più rapidamente possibile. Se i messaggi arrivano, vengono elaborati dal processo di commutazione.

Fase 2 - Elaborazione dei messaggi dalla nuova coda di trasmissione

Non appena non rimane alcun messaggio nella coda di trasmissione originale per il canale mittente del cluster, i nuovi messaggi vengono inseriti direttamente sulla nuova coda di trasmissione. Lo stato del canale indica che il canale mittente del cluster è associato alla nuova coda di trasmissione.

Il seguente messaggio viene scritto nel log degli errori gestore code: " AMQ7341 La coda di trasmissione per il canale *ChannelName* è *QueueName* ."

Attributi code di trasmissione cluster e code di trasmissione cluster multipli

È possibile inoltrare i messaggi del cluster a diversi gestori code che memorizzano i messaggi su una singola coda di trasmissione del cluster o su più code. Con una coda, si ha una serie di attributi della coda di trasmissione del cluster da impostare e interrogare; con più code, si hanno più serie. Per alcuni attributi, avere più insiemi è un vantaggio: ad esempio, l'interrogazione della profondità della coda indica quanti messaggi sono in attesa di essere inoltrati da uno o più canali, piuttosto che da tutti i canali. Per altri attributi, la presenza di più insiemi è uno svantaggio: ad esempio, probabilmente non si desidera configurare le stesse autorizzazioni di accesso per ogni coda di trasmissione del cluster. Per questo motivo, le autorizzazioni di accesso vengono sempre verificate rispetto al profilo per SYSTEM . CLUSTER . TRANSMIT . QUEUE e non rispetto ai profili per una particolare coda di trasmissione cluster. Se si desidera applicare controlli di sicurezza più granulari, consultare [Controllo accessi e code di trasmissione di più cluster](#).

Più canali mittente del cluster e più code di trasmissione

Un gestore code memorizza un messaggio su una coda di trasmissione cluster prima di inoltrarlo su un canale mittente del cluster. Seleziona un canale mittente del cluster connesso alla destinazione per il messaggio. Potrebbe avere una scelta di canali mittente del cluster che si connettono tutti alla stessa destinazione. La destinazione potrebbe essere la stessa coda fisica, connessa da più canali mittente del cluster a un singolo gestore code. La destinazione potrebbe essere anche molte code fisiche con lo stesso nome di coda, ospitate su gestori code differenti nello stesso cluster. Quando è disponibile una scelta di canali mittenti del cluster connessi a una destinazione, l'algoritmo di bilanciamento del carico di lavoro ne sceglie uno. La scelta dipende da una serie di fattori; consultare [L'algoritmo di gestione del carico di lavoro cluster](#).

In [Figura 38 a pagina 314](#), CL1 . QM1, CL1 . QM2 e CS . QM1 sono tutti canali che potrebbero portare alla stessa destinazione. Ad esempio, se si definisce Q1 in CL1 su QM1 e QM2 , CL1 . QM1 e CL1 . QM2 forniscono entrambi instradamenti alla stessa destinazione, Q1, su due gestori code differenti. Se il canale CS . QM1 si trova anche in CL1, è anche un canale che può essere utilizzato da un messaggio per Q1 . L'appartenenza al cluster di CS . QM1 potrebbe essere definita da un elenco nomi cluster, motivo per cui il nome del canale non include un nome cluster nella sua costruzione. In base ai parametri di bilanciamento del workload e all'applicazione di invio, alcuni messaggi per Q1 potrebbero essere posizionati su ognuna delle code di trasmissione, XMITQ . CL1 . QM1, XMITQ . CL1 e SYSTEM . CLUSTER . TRANSMIT . CS . QM1.

Se si intende separare il traffico dei messaggi, in modo che i messaggi per la stessa destinazione non condividano le code o i canali con i messaggi per destinazioni differenti, è necessario considerare prima come dividere il traffico su canali mittenti del cluster differenti e poi come separare i messaggi per un determinato canale su una coda di trasmissione differente. Le code cluster sullo stesso cluster, sullo stesso gestore code, normalmente condividono gli stessi canali cluster. La definizione di più code di

trasmissione del cluster da sola non è sufficiente per separare il traffico dei messaggi del cluster su code differenti. A meno che non si separano messaggi per code di destinazione differenti su canali differenti, i messaggi condividono la stessa coda di trasmissione cluster.

Un modo semplice per separare i canali utilizzati dai messaggi è creare più cluster. Su qualsiasi gestore code in ogni cluster, definire solo una coda cluster. Quindi, se si definisce un canale ricevente del cluster differente per ogni combinazione cluster / gestore code, i messaggi per ogni coda cluster non condividono un canale cluster con i messaggi per altre code cluster. Se si definiscono code di trasmissione separate per i canali cluster, il gestore code di invio memorizza i messaggi solo per una coda cluster su ciascuna coda di trasmissione. Ad esempio, se si desidera che due code cluster non condividano le risorse, è possibile posizionarle in cluster differenti sullo stesso gestore code o su gestori code differenti nello stesso cluster.

La scelta della coda di trasmissione cluster non influisce sull'algoritmo di bilanciamento del carico di lavoro. L'algoritmo di bilanciamento del workload sceglie quale canale mittente del cluster inoltrare un messaggio. Colloca il messaggio nella coda di trasmissione gestita da tale canale. Se l'algoritmo di bilanciamento del carico di lavoro viene richiamato per scegliere nuovamente, ad esempio se il canale si arresta, potrebbe essere in grado di selezionare un canale differente per inoltrare il messaggio. Se sceglie un canale differente e il nuovo canale inoltra i messaggi da una diversa coda di trasmissione del cluster, l'algoritmo di bilanciamento del carico di lavoro trasferisce il messaggio all'altra coda di trasmissione.

In [Figura 38 a pagina 314](#), due canali mittente del cluster, CS.QM1 e CS.QM2, sono associati alla coda di trasmissione del sistema predefinito. Quando l'algoritmo di bilanciamento del carico di lavoro memorizza un messaggio su SYSTEM.CLUSTER.TRANSMIT.QUEUE o qualsiasi altra coda di trasmissione del cluster, il nome del canale mittente del cluster che deve inoltrare il messaggio viene memorizzato nell'ID di correlazione del messaggio. Ogni canale inoltra solo i messaggi che corrispondono all'ID di correlazione con il nome canale.

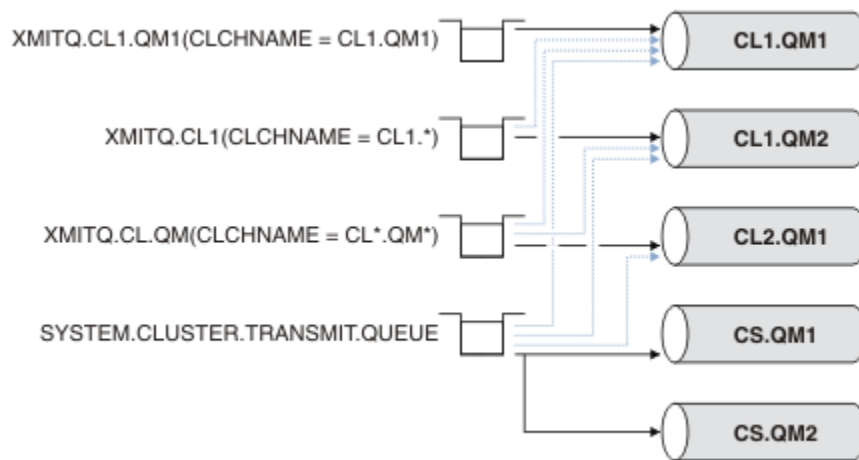


Figura 38. Più canali mittente cluster

Se CS.QM1 si arresta, vengono esaminati i messaggi sulla coda di trasmissione per quel canale mittente del cluster. I messaggi che possono essere inoltrati da un'altra canale vengono rielaborati dall'algoritmo di bilanciamento del carico di lavoro. Il loro ID di correlazione viene reimpostato su un nome canale mittente del cluster alternativo. Se il canale mittente del cluster alternativo è CS.QM2, il messaggio rimane su SYSTEM.CLUSTER.TRANSMIT.QUEUE. Se il canale alternativo è CL1.QM1, l'algoritmo di bilanciamento del workload trasferisce il messaggio a XMITQ.CL1.QM1. Quando il canale mittente del cluster viene riavviato, i nuovi messaggi e i messaggi che non sono stati contrassegnati per un canale mittente del cluster differente, vengono nuovamente trasferiti dal canale.

È possibile modificare l'associazione tra le code di trasmissione e i canali mittenti del cluster su un sistema in esecuzione. È possibile modificare un parametro **CLCHNAME** su una coda di trasmissione oppure modificare il parametro del gestore code **DEFCLXQ**. Quando un canale interessato dalla modifica viene riavviato, avvia il processo di commutazione della coda di trasmissione; consultare [“Come funziona il processo di commutazione del canale mittente del cluster in una coda di trasmissione differente” a pagina 312](#).

Il processo per commutare la coda di trasmissione inizia quando il canale viene riavviato. Il processo di ribilanciamento del carico di lavoro inizia quando il canale viene arrestato. I due processi possono essere eseguiti in parallelo.

Il semplice caso è quando l'arresto di un canale mittente del cluster non fa in modo che il processo di ribilanciamento modifichi il canale mittente del cluster che deve inoltrare i messaggi sulla coda. In questo caso, nessun altro canale mittente del cluster può inoltrare i messaggi alla destinazione corretta. Senza alcun canale mittente del cluster alternativo per inoltrare i messaggi alla relativa destinazione, i messaggi rimangono contrassegnati per lo stesso canale mittente del cluster dopo l'arresto del canale mittente del cluster. Quando il canale viene avviato, se uno switch è in sospenso, il processo di commutazione sposta i messaggi in una coda di trasmissione differente in cui vengono elaborati dallo stesso canale mittente del cluster.

Il caso più complesso è quello in cui più di un canale mittente del cluster può elaborare alcuni messaggi nella stessa destinazione. Arrestare e riavviare il canale mittente del cluster per attivare lo switch della coda di trasmissione. In molti casi, al momento del riavvio del canale, l'algoritmo di bilanciamento del carico di lavoro ha già spostato i messaggi dalla coda di trasmissione originale a code di trasmissione differenti servite da canali mittenti del cluster differenti. Solo i messaggi che non possono essere inoltrati da un diverso canale mittente del cluster rimangono da trasferire alla nuova coda di trasmissione. In alcuni casi, se il canale viene riavviato rapidamente, rimangono alcuni messaggi che potrebbero essere trasferiti dall'algoritmo di bilanciamento del carico di lavoro. In tal caso, alcuni messaggi rimanenti vengono commutati dal processo di bilanciamento del workload e altri dal processo di commutazione della coda di trasmissione.

Concetti correlati

Canali cluster

Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster

“Calcolo della dimensione del log” a pagina 671

Stima della dimensione del log necessaria per un gestore code.

Attività correlate

Clustering: pianificazione della configurazione delle code di trasmissione del cluster

“Creazione di due cluster sovrapposti con un gestore code del gateway” a pagina 352

Seguire le istruzioni nell'attività per creare cluster sovrapposti con un gestore code del gateway. Utilizzare i cluster come punto iniziale per i seguenti esempi di isolamento dei messaggi in un'applicazione da messaggi in altre applicazioni in un cluster.

“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

“Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 359

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una coda di trasmissione cluster aggiuntiva per separare il traffico di messaggi a un singolo gestore code in un cluster.

“Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 362

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza un cluster aggiuntivo per isolare i messaggi in una particolare coda cluster.

Come stabilire la comunicazione in un cluster

Un iniziatore di canale è necessario per avviare un canale di comunicazione quando è presente un messaggio da consegnare. Un listener del canale attende di avviare l'altra estremità di un canale per ricevere il messaggio.

Prima di iniziare

Per stabilire la comunicazione tra i gestori code in un cluster, configurare un collegamento utilizzando uno dei protocolli di comunicazione supportati. I protocolli supportati sono:

- TCP o LU 6.2 su qualsiasi piattaforma
- **Windows** NetBIOS o SPX su sistemi Windows

Come parte di questa configurazione, hai anche bisogno degli iniziatori di canali e dei listener di canali proprio come si fa con l'accodamento distribuito.

Informazioni su questa attività

Tutti i gestori code cluster hanno bisogno di un iniziatore di canali per monitorare la coda di iniziazione definita dal sistema `SYSTEM.CHANNEL.INITQ`. `SYSTEM.CHANNEL.INITQ` è la coda di iniziazione per tutte le code di trasmissione, inclusa la coda di trasmissione cluster.

Ogni gestore code deve avere un listener del canale. Un programma del listener del canale attende le richieste di rete in entrata e avvia il canale ricevente appropriato quando è necessario. L'implementazione dei listener del canale è specifica della piattaforma, tuttavia ci sono alcune caratteristiche comuni.

Su tutte le piattaforme IBM MQ, il listener può essere avviato utilizzando il comando **START LISTENER**.

Multi Su Multiplatforms, è possibile avviare il listener automaticamente contemporaneamente al gestore code. Per avviare automaticamente il listener, impostare l'attributo `CONTROL` dell'oggetto `LISTENER` su `QMGR` o `STARTONLY`.

z/OS Una porta listener non condivisa (`INDISP(QMGR)`) deve essere utilizzata per canali `CLUSRCVR` su `z/OS` e per canali `CLUSDR` su `z/OS`.

Procedura

1. Avviare l'iniziatore di canali.

- **z/OS** Su `z/OS`, esiste un iniziatore di canali per ogni gestore code e viene eseguito come uno spazio di indirizzo separato. Avviarlo utilizzando il comando **MQSC START CHINIT**, che viene immesso come parte dell'avvio del gestore code.
- **ALW** Su AIX, Linux, and Windows, quando si avvia un gestore code, se l'attributo del gestore code `SCHINIT` è impostato su `QMGR`, viene avviato automaticamente un iniziatore di canali. Altrimenti, può essere avviato utilizzando il comando **runmqsc START CHINIT** o il comando di controllo **runmqchi**.
- **IBM i** Su IBM i, quando si avvia un gestore code, se l'attributo del gestore code `SCHINIT` è impostato su `QMGR`, viene avviato automaticamente un iniziatore di canali. Altrimenti, può essere avviato utilizzando il comando **runmqsc START CHINIT** o il comando di controllo **runmqchi**.

2. Avviare il listener del canale.

- **z/OS** Su `z/OS`, utilizzare il programma listener del canale fornito da IBM MQ. Per avviare un listener del canale IBM MQ, utilizzare il **MQSC** comando `START LISTENER`, che viene emesso come parte dell'avvio dell'iniziatore del canale. Ad esempio:

```
START LISTENER PORT(1414) TRPTYPE(TCP)
```

o:

```
START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
```

I membri di un gruppo di condivisione code possono utilizzare un listener condiviso invece di un listener per ogni gestore code. Non utilizzare listener condivisi con i cluster. In particolare, non rendere il CONNAME del canale CLUSRCVR l'indirizzo del listener condiviso del gruppo di condivisione code. In tal caso, i gestori code potrebbero ricevere messaggi per le code per le quali non dispongono di una definizione.

- **IBM i** Su IBM i, utilizzare il programma listener del canale fornito da IBM MQ. Per avviare un listener del canale IBM MQ utilizzare il comando **CL STRMQLSR**. Ad esempio:

```
STRMQLSR MQMNAME(QM1) PORT(1414)
```

- **Windows** Su Windows, utilizzare il programma listener del canale fornito da IBM MQo le funzioni fornite dal sistema operativo.

Per avviare il listener del canale IBM MQ utilizzare il comando **RUNMQLSR**. Ad esempio:

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

- **Linux** **AIX** Su AIX and Linux, utilizzare il programma listener del canale fornito da IBM MQo le funzionalità fornite dal sistema operativo; ad esempio, **inetd** per le comunicazioni TCP.

Per avviare il listener del canale IBM MQ utilizzare il comando **runmqlsr**. Ad esempio:

```
runmqlsr -t tcp -p 1414 -m QM1
```

Per utilizzare **inetd** per avviare i canali, configurare due file:

- a. Modificare il file `/etc/services`. È necessario essere collegati come superuser o root. Se la seguente riga non è nel file, aggiungerla come mostrato:

```
MQSeries 1414/tcp # WebSphere MQ channel listener
```

dove 1414 è il numero di porta richiesto da IBM MQ. È possibile modificare il numero di porta, ma deve corrispondere al numero di porta specificato all'estremità di invio.

- b. Modificare il file `/etc/inetd.conf`. Se non si dispone della seguente riga in tale file, aggiungerla come mostrato:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
-m queue.manager.name
```

dove `MQ_INSTALLATION_PATH` viene sostituito dalla directory di alto livello in cui è installato IBM MQ.

Gli aggiornamenti diventano attivi dopo che **inetd** ha riletto i file di configurazione. Immettere i comandi riportati di seguito dall'ID utente root:

AIX Su AIX:

```
refresh -s inetd
```

Linux Su Linux:

- a. Individuare l'ID processo di **inetd** con il seguente comando:

```
ps -ef | grep inetd
```

b. Eseguire il comando appropriato.

Per Linux:

```
kill -1 inetd processid
```

Configurazione di un nuovo cluster

Seguire queste istruzioni per configurare il cluster di esempio. Istruzioni separate descrivono l'impostazione del cluster su TCP/IP, LU 6.2e con una o più code di trasmissione. Verificare il funzionamento del cluster inviando un messaggio da un gestore code all'altro.

Prima di iniziare

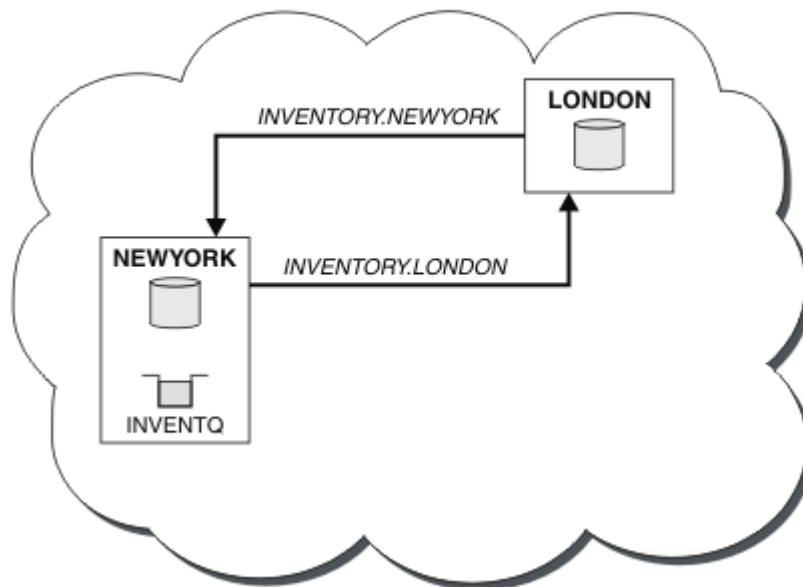
- Invece di attenersi alle seguenti istruzioni, è possibile utilizzare una delle procedure guidate fornite con IBM MQ Explorer per creare un cluster come quello creato da questa attività. Fare clic con il tastino destro del mouse sulla cartella Cluster di gestori code, quindi fare clic su **Nuovo > Cluster di gestori code** e seguire le istruzioni fornite nella procedura guidata.
- Per informazioni di background che aiutino l'utente a comprendere le operazioni eseguite per configurare un cluster, consultare [“Definizione di code cluster”](#) a pagina 304, [Canali cluster](#) e [Listener](#).

Informazioni su questa attività

Si sta configurando una nuova rete IBM MQ per un chain store. Il negozio ha due filiali, una a Londra e una a New York. I dati e le applicazioni per ogni negozio sono ospitati da sistemi che eseguono gestori code separati. I due gestori code sono denominati LONDON e NEWYORK. L'applicazione di inventario viene eseguita sul sistema a New York, connesso al gestore code NEWYORK. L'applicazione è guidata dall'arrivo dei messaggi sulla coda INVENTQ, ospitata da NEWYORK. I due gestori code, LONDON e NEWYORK, devono essere collegati in un cluster denominato INVENTORY in modo che possano entrambi inserire messaggi in INVENTQ.

Questo è il seguente aspetto del

INVENTORY




cluster:

È possibile configurare ciascun gestore code nel cluster per inviare messaggi ad altri gestori code nel cluster utilizzando code di trasmissione del cluster differenti.

Le istruzioni per impostare il cluster variano in base al protocollo di trasporto, al numero di code di trasmissione o alla piattaforma. Hai una scelta di tre combinazioni. La procedura di verifica rimane la stessa per tutte le combinazioni.

INVENTORY è un cluster piccolo. Tuttavia, è utile come prova di concetto. La cosa importante da capire su questo cluster è l'ambito che offre per il miglioramento futuro.

Procedura

- [“Impostazione di un cluster utilizzando TCP/IP con una coda di trasmissione singola per gestore code” a pagina 319](#)
- [“Configurazione di un cluster su TCP/IP utilizzando più code di trasmissione per gestore code” a pagina 322](#)
-  [“Configurazione di un cluster utilizzando LU 6.2 su z/OS” a pagina 325](#)
- [“Verifica del cluster” a pagina 327](#)

Concetti correlati

[Cluster](#)

[Confronto tra cluster e accodamento distribuito](#)

[Componenti di un cluster](#)

Attività correlate

[“Configurazione di un cluster di gestore code” a pagina 303](#)

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. È possibile definire componenti cluster e creare e gestire cluster.

Impostazione di un cluster utilizzando TCP/IP con una coda di trasmissione singola per gestore code


Questo è uno dei tre argomenti che descrivono diverse configurazioni per un cluster semplice.

Prima di iniziare

Per una panoramica del cluster che viene creato, consultare [“Configurazione di un nuovo cluster” a pagina 318](#).

L'attributo del gestore code, **DEFCLXQ**, deve essere lasciato come valore predefinito, SCTQ.

Informazioni su questa attività

Attenersi a questa procedura per impostare un cluster su [Multiplatforme](#) utilizzando il protocollo di trasporto TCP/IP.  Su z/OS, è necessario seguire le istruzioni in [“Defining a TCP connection on z/OS” a pagina 982](#) per impostare la connessione TCP/IP, anziché definire il listener nel passo [“4” a pagina 320](#). Altrimenti, i passaggi sono gli stessi per z/OS, ma i messaggi di errore vengono scritti nella console, piuttosto che nel log degli errori del gestore code.

Procedura

1. Decidere l'organizzazione del cluster e il relativo nome.

Si è deciso di collegare i due gestori code, LONDON e NEWYORK, in un cluster. Un cluster con solo due gestori code offre solo vantaggi marginali su una rete che utilizza l'accodamento distribuito. Si tratta di un buon modo per iniziare e offre un margine di espansione futura. Quando si aprono nuove filiali del negozio, è possibile aggiungere facilmente i gestori code al cluster. L'aggiunta di nuovi gestori code non interrompe la rete esistente; consultare [“Aggiunta di un gestore code a un cluster” a pagina 329](#).

Per il momento, l'unica applicazione in esecuzione è l'applicazione di inventario. Il nome cluster è INVENTORY.

2. Decidere quali gestori code devono contenere repository completi.

In qualsiasi cluster è necessario denominare almeno un gestore code, o preferibilmente due, per conservare i repository completi. In questo esempio, ci sono solo due gestori code, LONDON e NEWYORK, che contengono repository completi.

- a. È possibile eseguire i passi rimanenti in qualsiasi ordine.
- b. Man mano che si procede con la procedura, i messaggi di avvertenza potrebbero essere scritti nel log del gestore code. I messaggi sono il risultato di definizioni mancanti che devono essere ancora aggiunte.

```
Examples of the responses to the commands are shown in a box
like this after each step in this task.
These examples show the responses returned by IBM MQ for AIX.
The responses vary on other platforms.
```

- c. Prima di procedere con questi passi, assicurarsi che i gestori code siano avviati.

3. Modificare le definizioni del gestore code per aggiungere le definizioni del repository.

Su ogni gestore code che deve contenere un repository completo, modificare la definizione di gestore code locale utilizzando il comando ALTER QMGR e specificando l'attributo REPOS :

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

Ad esempio, se si immette:

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON viene modificato in repository completo.

4. Definire i listener.

Definire un listener che accetti richieste di rete da altri gestori code per ogni gestore code nel cluster. Sui gestori code LONDON , immettere il seguente comando:

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

L'attributo CONTROL assicura che il listener venga avviato e arrestato quando il gestore code lo fa.

Il listener non viene avviato quando è definito, quindi deve essere avviato manualmente la prima volta con il seguente comando MQSC:

```
START LISTENER(LONDON_LS)
```

Immettere comandi simili per tutti gli altri gestori code nel cluster, modificando il nome listener per ciascun gestore code.

Esistono diversi modi per definire questi listener, come mostrato in [Listener](#).

5. Definire il canale CLUSRCVR per il gestore code LONDON .

Su ogni gestore code in un cluster, è possibile definire un canale ricevente del cluster su cui il gestore code può ricevere messaggi. Vedere [Cluster - receiver channel: CLUSRCVR](#) . Il canale CLUSRCVR definisce il nome della connessione del gestore code. Il nome della connessione è memorizzato nei


repository, a cui possono fare riferimento altri gestori code. La parola chiave CLUSTER mostra la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster.

In questo esempio, il nome del canale è INVENTORY.LONDON e il nome della connessione (CONNAME) è l'indirizzo di rete della macchina su cui si trova il gestore code, ovvero LONDON.CHSTORE.COM. L'indirizzo di rete può essere immesso come un nome host DNS alfanumerico o come un indirizzo IP in formato decimale con punti IPv4. Ad esempio, 192.0.2.0 o IPv6 formato esadecimale; ad esempio 2001:DB8:0204:acff:fe97:2c34:fde0:3485. Il numero di porta non è specificato, quindi viene utilizzata la porta predefinita (1414).

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. Definire il canale CLUSRCVR per il gestore code NEWYORK .

 Se il listener del canale utilizza la porta predefinita, generalmente 1414, e il cluster non include un gestore code su z/OS, è possibile omettere CONNAME.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

7. Definire il canale CLUSSDR sul gestore code LONDON .

Definire manualmente il canale CLUSSDR da ogni gestore code del repository completo a ogni altro gestore code del repository completo nel cluster. Fare riferimento a [Canale mittente del cluster: CLUSSDR](#). In questo caso, ci sono solo due gestori code, entrambi con repository completi. Ciascuno di essi ha bisogno di un canale CLUSSDR definito manualmente che punti al canale CLUSRCVR definito sull'altro gestore code. I nomi canale forniti nelle definizioni CLUSSDR devono corrispondere ai nomi canale nelle definizioni CLUSRCVR corrispondenti. Quando un gestore code dispone di definizioni sia per un canale ricevente del cluster che per un canale mittente del cluster nello stesso cluster, il canale mittente del cluster viene avviato.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. Definire il canale CLUSSDR sul gestore code NEWYORK .

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. Definire la coda cluster INVENTQ

Definire la coda INVENTQ sul gestore code NEWYORK, specificando la parola chiave CLUSTER.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

La parola chiave CLUSTER fa sì che la coda venga annunciata al cluster. Una volta definita, la coda diventa disponibile per gli altri gestori code nel cluster. Possono inviargli messaggi senza dover creare una definizione di coda remota.

Tutte le definizioni sono complete. Su tutte le piattaforme, avviare un programma listener su ciascun gestore code. Il programma listener attende le richieste di rete in arrivo e avvia il canale ricevente del cluster quando è necessario.

Operazioni successive

Sei ora pronto a [verificare il cluster](#).

Attività correlate

[“Configurazione di un cluster su TCP/IP utilizzando più code di trasmissione per gestore code”](#) a pagina 322

Questo è uno dei tre argomenti che descrivono diverse configurazioni per un cluster semplice.

[“Configurazione di un cluster utilizzando LU 6.2 su z/OS”](#) a pagina 325

Questo è uno degli argomenti della struttura ad albero che descrivono diverse configurazioni per un cluster semplice.

Configurazione di un cluster su TCP/IP utilizzando più code di trasmissione per gestore code

Questo è uno dei tre argomenti che descrivono diverse configurazioni per un cluster semplice.

Prima di iniziare

Per una panoramica del cluster che viene creato, consultare [“Configurazione di un nuovo cluster”](#) a pagina 318.

Informazioni su questa attività

Attenersi a questa procedura per impostare un cluster su [Multiplatforme](#) utilizzando il protocollo di trasporto TCP/IP. I gestori code del repository sono configurati per utilizzare una coda di trasmissione cluster differente per inviare messaggi l'uno all'altro e ad altri gestori code nel cluster. Se si aggiungono gestori code al cluster che devono utilizzare anche code di trasmissione differenti, seguire l'attività, [“Aggiunta di un gestore code a un cluster: code di trasmissione separate”](#) a pagina 331.

Procedura

1. Decidere l'organizzazione del cluster e il relativo nome.

Si è deciso di collegare i due gestori code, LONDON e NEWYORK, in un cluster. Un cluster con solo due gestori code offre solo vantaggi marginali su una rete che utilizza l'accodamento distribuito. Si tratta di un buon modo per iniziare e offre un margine di espansione futura. Quando si aprono nuove filiali del negozio, è possibile aggiungere facilmente i gestori code al cluster. L'aggiunta di nuovi gestori code non interrompe la rete esistente; consultare [“Aggiunta di un gestore code a un cluster”](#) a pagina 329.

Per il momento, l'unica applicazione in esecuzione è l'applicazione di inventario. Il nome cluster è INVENTORY.

2. Decidere quali gestori code devono contenere repository completi.

In qualsiasi cluster è necessario denominare almeno un gestore code, o preferibilmente due, per conservare i repository completi. In questo esempio, ci sono solo due gestori code, LONDON e NEWYORK, che contengono repository completi.

- a. È possibile eseguire i passi rimanenti in qualsiasi ordine.

- b. Man mano che si procede con la procedura, i messaggi di avvertenza potrebbero essere scritti nel log del gestore code. I messaggi sono il risultato di definizioni mancanti che devono essere ancora aggiunte.

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by IBM MQ for AIX. The responses vary on other platforms.

- c. Prima di procedere con questi passi, assicurarsi che i gestori code siano avviati.
3. Modificare le definizioni del gestore code per aggiungere le definizioni del repository.

Su ogni gestore code che deve contenere un repository completo, modificare la definizione di gestore code locale utilizzando il comando ALTER QMGR e specificando l'attributo REPOS :

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

Ad esempio, se si immette:

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON viene modificato in repository completo.

4. Modificare le definizioni del gestore code per creare code di trasmissione cluster separate per ciascuna destinazione.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Su ogni gestore code che si aggiunge al cluster decidere se utilizzare o meno code di trasmissione separate. Consultare gli argomenti [“Aggiunta di un gestore code a un cluster”](#) a pagina 329 e [“Aggiunta di un gestore code a un cluster: code di trasmissione separate”](#) a pagina 331.

5. Definire i listener.

Definire un listener che accetti richieste di rete da altri gestori code per ogni gestore code nel cluster. Sui gestori code LONDON , immettere il seguente comando:

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

L'attributo CONTROL assicura che il listener venga avviato e arrestato quando il gestore code lo fa.

Il listener non viene avviato quando è definito, quindi deve essere avviato manualmente la prima volta con il seguente comando MQSC:

```
START LISTENER(LONDON_LS)
```

Immettere comandi simili per tutti gli altri gestori code nel cluster, modificando il nome listener per ciascun gestore code.

Esistono diversi modi per definire questi listener, come mostrato in [Listener](#).

6. Definire il canale CLUSRCVR per il gestore code LONDON .

Su ogni gestore code in un cluster, è possibile definire un canale ricevente del cluster su cui il gestore code può ricevere messaggi. Vedere [Cluster - receiver channel: CLUSRCVR](#) . Il canale CLUSRCVR definisce il nome della connessione del gestore code. Il nome della connessione è memorizzato nei


repository, a cui possono fare riferimento altri gestori code. La parola chiave CLUSTER mostra la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster.

In questo esempio, il nome del canale è INVENTORY.LONDON e il nome della connessione (CONNAME) è l'indirizzo di rete della macchina su cui si trova il gestore code, ovvero LONDON.CHSTORE.COM. L'indirizzo di rete può essere immesso come un nome host DNS alfanumerico o come un indirizzo IP in formato decimale con punti IPv4. Ad esempio, 192.0.2.0o IPv6 formato esadecimale; ad esempio 2001:DB8:0204:acff:fe97:2c34:fde0:3485. Il numero di porta non è specificato, quindi viene utilizzata la porta predefinita (1414).

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. Definire il canale CLUSRCVR per il gestore code NEWYORK.

 Se il listener del canale utilizza la porta predefinita, generalmente 1414, e il cluster non include un gestore code su z/OS, è possibile omettere CONNAME.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

8. Definire il canale CLUSSDR sul gestore code LONDON.

Definire manualmente il canale CLUSSDR da ogni gestore code del repository completo a ogni altro gestore code del repository completo nel cluster. Fare riferimento a [Canale mittente del cluster: CLUSSDR](#). In questo caso, ci sono solo due gestori code, entrambi con repository completi. Ciascuno di essi ha bisogno di un canale CLUSSDR definito manualmente che punti al canale CLUSRCVR definito sull'altro gestore code. I nomi canale forniti nelle definizioni CLUSSDR devono corrispondere ai nomi canale nelle definizioni CLUSRCVR corrispondenti. Quando un gestore code dispone di definizioni sia per un canale ricevente del cluster che per un canale mittente del cluster nello stesso cluster, il canale mittente del cluster viene avviato.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

9. Definire il canale CLUSSDR sul gestore code NEWYORK.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

10. Definire la coda cluster INVENTQ

Definire la coda INVENTQ sul gestore code NEWYORK, specificando la parola chiave CLUSTER.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

La parola chiave CLUSTER fa sì che la coda venga annunciata al cluster. Una volta definita, la coda diventa disponibile per gli altri gestori code nel cluster. Possono inviargli messaggi senza dover creare una definizione di coda remota.

Tutte le definizioni sono complete. Su tutte le piattaforme, avviare un programma listener su ciascun gestore code. Il programma listener attende le richieste di rete in arrivo e avvia il canale ricevente del cluster quando è necessario.

Operazioni successive

Sei ora pronto a verificare il cluster.

Attività correlate

[“Impostazione di un cluster utilizzando TCP/IP con una coda di trasmissione singola per gestore code” a pagina 319](#)

Questo è uno dei tre argomenti che descrivono diverse configurazioni per un cluster semplice.

[“Configurazione di un cluster utilizzando LU 6.2 su z/OS” a pagina 325](#)

Questo è uno degli argomenti della struttura ad albero che descrivono diverse configurazioni per un cluster semplice.

Configurazione di un cluster utilizzando LU 6.2 su z/OS

Questo è uno degli argomenti della struttura ad albero che descrivono diverse configurazioni per un cluster semplice.

Prima di iniziare

Per una panoramica del cluster che viene creato, consultare [“Configurazione di un nuovo cluster” a pagina 318](#).

Procedura

1. Decidere l'organizzazione del cluster e il relativo nome.

Si è deciso di collegare i due gestori code, LONDON e NEWYORK, in un cluster. Un cluster con solo due gestori code offre solo vantaggi marginali su una rete che utilizza l'accodamento distribuito. Si tratta di un buon modo per iniziare e offre un margine di espansione futura. Quando si aprono nuove filiali del negozio, è possibile aggiungere facilmente i gestori code al cluster. L'aggiunta di nuovi gestori code non interrompe la rete esistente; consultare [“Aggiunta di un gestore code a un cluster” a pagina 329](#).

Per il momento, l'unica applicazione in esecuzione è l'applicazione di inventario. Il nome cluster è INVENTORY.

2. Decidere quali gestori code devono contenere repository completi.

In qualsiasi cluster è necessario denominare almeno un gestore code, o preferibilmente due, per conservare i repository completi. In questo esempio, ci sono solo due gestori code, LONDON e NEWYORK, che contengono repository completi.

- a. È possibile eseguire i passi rimanenti in qualsiasi ordine.
 - b. Man mano che si procede con i passi, potrebbero essere scritti messaggi di avvertenza nella console del sistema z/OS. I messaggi sono il risultato di definizioni mancanti che devono essere ancora aggiunte.
 - c. Prima di procedere con questi passi, assicurarsi che i gestori code siano avviati.
3. Modificare le definizioni del gestore code per aggiungere le definizioni del repository.

Su ogni gestore code che deve contenere un repository completo, modificare la definizione di gestore code locale utilizzando il comando ALTER QMGR e specificando l'attributo REPOS :

```
ALTER QMGR REPOS(INVENTORY)
```


```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

Ad esempio, se si immette:

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON viene modificato in repository completo.

4. Definire i listener.

 Consultare [L'iniziatore di canali su z/OS e "Receiving on LU 6.2"](#) a pagina 986.

Il listener non viene avviato quando è definito, quindi deve essere avviato manualmente la prima volta con il seguente comando MQSC:

```
START LISTENER(LONDON_LS)
```

Immettere comandi simili per tutti gli altri gestori code nel cluster, modificando il nome listener per ciascun gestore code.

5. Definire il canale CLUSRCVR per il gestore code LONDON .

Su ogni gestore code in un cluster, è possibile definire un canale ricevente del cluster su cui il gestore code può ricevere messaggi. Vedere [Cluster - receiver channel: CLUSRCVR](#) . Il canale CLUSRCVR definisce il nome della connessione del gestore code. Il nome della connessione è memorizzato nei repository, a cui possono fare riferimento altri gestori code. La parola chiave CLUSTER mostra la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. Definire il canale CLUSRCVR per il gestore code NEWYORK .

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

7. Definire il canale CLUSSDR sul gestore code LONDON .

Definire manualmente il canale CLUSSDR da ogni gestore code del repository completo a ogni altro gestore code del repository completo nel cluster. Fare riferimento a [Canale mittente del cluster: CLUSSDR](#) . In questo caso, ci sono solo due gestori code, entrambi con repository completi. Ciascuno di essi ha bisogno di un canale CLUSSDR definito manualmente che punti al canale CLUSRCVR definito sull'altro gestore code. I nomi canale forniti nelle definizioni CLUSSDR devono corrispondere ai nomi canale nelle definizioni CLUSRCVR corrispondenti. Quando un gestore code dispone di definizioni sia

per un canale ricevente del cluster che per un canale mittente del cluster nello stesso cluster, il canale mittente del cluster viene avviato.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. Definire il canale CLUSSDR sul gestore code NEWYORK .

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. Definire la coda cluster INVENTQ

Definire la coda INVENTQ sul gestore code NEWYORK , specificando la parola chiave CLUSTER .

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

La parola chiave CLUSTER fa sì che la coda venga annunciata al cluster. Una volta definita, la coda diventa disponibile per gli altri gestori code nel cluster. Possono inviargli messaggi senza dover creare una definizione di coda remota.

Tutte le definizioni sono complete. Su tutte le piattaforme, avviare un programma listener su ciascun gestore code. Il programma listener attende le richieste di rete in arrivo e avvia il canale ricevente del cluster quando è necessario.

Operazioni successive

Sei ora pronto a [verificare il cluster](#).

Attività correlate

[“Impostazione di un cluster utilizzando TCP/IP con una coda di trasmissione singola per gestore code” a pagina 319](#)

Questo è uno dei tre argomenti che descrivono diverse configurazioni per un cluster semplice.

[“Configurazione di un cluster su TCP/IP utilizzando più code di trasmissione per gestore code” a pagina 322](#)

Questo è uno dei tre argomenti che descrivono diverse configurazioni per un cluster semplice.


Verifica del cluster

Gli argomenti peer descrivono tre diverse configurazioni per un cluster semplice. Questo argomento spiega come verificare il cluster.

Prima di iniziare

In questo argomento si presuppone che si stia verificando un cluster creato mediante una delle seguenti attività:

- [“Impostazione di un cluster utilizzando TCP/IP con una coda di trasmissione singola per gestore code” a pagina 319](#).

- [“Configurazione di un cluster su TCP/IP utilizzando più code di trasmissione per gestore code”](#) a pagina 322.
-  [“Configurazione di un cluster utilizzando LU 6.2 su z/OS”](#) a pagina 325.

Per una panoramica del cluster creato, consultare [“Configurazione di un nuovo cluster”](#) a pagina 318.

Informazioni su questa attività

È possibile verificare il cluster in uno o più dei seguenti modi:

1. Esecuzione dei comandi di gestione per visualizzare gli attributi del cluster e del canale.
2. Eseguire i programmi di esempio per inviare e ricevere messaggi su una coda cluster.
3. Scrivere i propri programmi per inviare un messaggio di richiesta ad una coda cluster e rispondere con un messaggio di risposta ad una coda di risposta non cluster.

Procedura

Immettere i comandi `DISPLAY runmqsc` per verificare il cluster.

Le risposte che si vedono dovrebbero essere come le risposte nei passi che seguono.

1. Dal gestore code NEWYORK , eseguire il comando **DISPLAY CLUSQMGR** :

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)      CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)      CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. Dal gestore code NEWYORK , eseguire il comando **DISPLAY CHANNEL STATUS** :

```
dis chstatus(*)
```

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK) XMITQ( )
CONNNAME(192.0.2.0)        CURRENT
CHLTYPE(CLUSRCVR)         STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNNAME(192.0.2.1)        CURRENT
CHLTYPE(CLUSSDR)          STATUS(RUNNING)
RQMNAME(LONDON)
```

Inviare messaggi tra i due gestori code, utilizzando **amqsput**.

3. Su LONDON eseguire il comando **amqsput INVENTQ LONDON**.

Immettere alcuni messaggi, seguiti da una riga vuota.

4. Su NEWYORK eseguire il comando **amqsget INVENTQ NEWYORK**.

Ora vengono visualizzati i messaggi immessi su LONDON. Dopo 15 secondi il programma termina.

Inviare i messaggi tra i due gestori code utilizzando i propri programmi.

Nei seguenti passi, LONDON inserisce un messaggio in INVENTQ alle NEWYORK e riceve una risposta sulla coda LONDON_reply.

5. Su LONDON , inserire un messaggio nella coda cluster.

- a) Definire una coda locale denominata LONDON_reply.
 - b) Impostare le opzioni MQOPEN su MQOO_OUTPUT.
 - c) Emettere la chiamata MQOPEN per aprire la coda INVENTQ.
 - d) Impostare il nome *ReplyToQ* nel descrittore del messaggio su LONDON_reply.
 - e) Emettere la chiamata MQPUT per inserire il messaggio.
 - f) Eseguire il commit del messaggio.
6. Su NEWYORK , ricevere il messaggio sulla coda cluster e inserire una risposta nella coda di risposta.
- a) Impostare le opzioni MQOPEN su MQOO_BROWSE.
 - b) Emettere la chiamata MQOPEN per aprire la coda INVENTQ.
 - c) Emettere la chiamata MQGET per ottenere un messaggio da INVENTQ.
 - d) Richiamare il nome *ReplyToQ* dal descrittore del messaggio.
 - e) Inserire il nome *ReplyToQ* nel campo `ObjectName` del descrittore oggetto.
 - f) Impostare le opzioni MQOPEN su MQOO_OUTPUT.
 - g) Emettere la chiamata MQOPEN per aprire LONDON_reply sul gestore code LONDON.
 - h) Emettere la chiamata MQPUT per inserire il messaggio in LONDON_reply.
7. Su LONDON ricevere la risposta.
- a) Impostare le opzioni MQOPEN su MQOO_BROWSE.
 - b) Emettere la chiamata MQOPEN per aprire la coda LONDON_reply.
 - c) Emettere la chiamata MQGET per richiamare il messaggio da LONDON_reply.

Aggiunta di un gestore code a un cluster

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code e gli argomenti del cluster vengono trasferiti utilizzando la singola coda di trasmissione del cluster `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è configurato come descritto in [“Configurazione di un nuovo cluster”](#) a pagina 318. Contiene due gestori code, LONDON e NEWYORK, che contengono entrambi repository completi.
- Il gestore code PARIS appartiene all'installazione primaria. Se non lo è, è necessario eseguire il comando `setmqenv` per impostare l'ambiente di comandi per l'installazione a cui appartiene PARIS .
- La connettività TCP esiste tra tutti e tre i sistemi e il gestore code è configurato con un listener TCP che viene avviato sotto il controllo del gestore code.

Informazioni su questa attività

1. Una nuova filiale della catena di negozi è in fase di configurazione a Parigi e si desidera aggiungere al cluster un gestore code denominato PARIS .
2. Il gestore code PARIS invia aggiornamenti di inventario all'applicazione in esecuzione sul sistema a New York inserendo i messaggi nella coda INVENTQ .

Attendersi alla seguente procedura per aggiungere un gestore code a un cluster.

Procedura

1. Decidere quale repository completo PARIS fa riferimento per primo.

Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi. Raccoglie le informazioni sul cluster da un repository completo e quindi crea il proprio repository parziale. Scegliere uno dei repository come repository completo. Non appena un nuovo gestore code viene aggiunto al cluster, viene immediatamente a conoscenza dell'altro repository. Le informazioni relative alle modifiche a un gestore code vengono inviate direttamente a due repository. In questo esempio, si collega PARIS al gestore code LONDON, solo per motivi geografici.

Nota: Eseguire i passi rimanenti in qualsiasi ordine, dopo l'avvio del gestore code PARIS .

2. Definire un canale CLUSRCVR sul gestore code PARIS.

Ogni gestore code in un cluster deve definire un canale ricevente del cluster su cui può ricevere i messaggi. Su PARIS, definire:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

Il canale ricevente del cluster annuncia la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster INVENTORY. Non creare definizioni su altri gestori code per l'invio al canale ricevente del cluster INVENTORY . PARIS. Le altre definizioni vengono effettuate automaticamente quando necessario. Vedere [Canali cluster](#).

3. z/OS

Avviare l'inziatore di canali su IBM MQ for z/OS.

4. Definire un canale CLUSSDR nel gestore code PARIS.

Quando si aggiunge a un cluster un gestore code che non è un repository completo, si definisce un solo canale mittente del cluster per stabilire una connessione iniziale a tale repository. Fare riferimento a [Canale mittente del cluster: CLUSSDR](#).

Su PARIS, creare la seguente definizione per un canale CLUSSDR denominato INVENTORY . LONDON nel gestore code con indirizzo di rete LONDON . CHSTORE . COM.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

5. Opzionale: Se si sta aggiungendo a un cluster un gestore code che è stato precedentemente rimosso dallo stesso cluster, verificare che venga visualizzato come membro del cluster. In caso contrario, completare i seguenti passi aggiuntivi:

a) Immettere il comando **REFRESH CLUSTER** sul gestore code che si sta aggiungendo.

Questa fase arresta i canali del cluster e fornisce alla tua cache del cluster locale una nuova serie di numeri di sequenza che sono sicuri di essere aggiornati nel resto del cluster.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

Nota: Per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può danneggiare il cluster mentre è in esecuzione e, di nuovo, a intervalli di 27 giorni, quando gli oggetti del cluster inviano automaticamente aggiornamenti sullo stato a tutti i gestori code interessati. Consultare [Refreshing in a large cluster can affect performance and availability of the cluster](#).

b) Riavvia il canale CLUSSDR

(ad esempio, utilizzando il comando [START CHANNEL](#)).

c) Riavviare il canale CLUSRCVR.

Risultati

La seguente figura mostra il cluster configurato da questa attività.

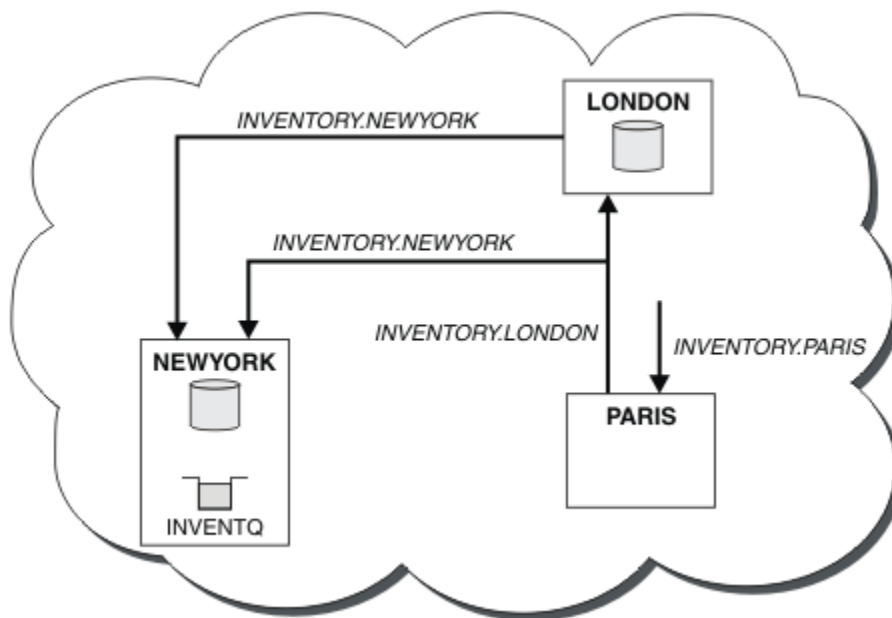


Figura 39. Il cluster INVENTORY con tre gestori code

Creando solo due definizioni, una CLUSRCVR e una CLUSSDR , è stato aggiunto il gestore code PARIS al cluster.

Ora il gestore code PARIS apprende, dal repository completo in LONDON, che la coda INVENTQ è ospitata dal gestore code NEWYORK. Quando un'applicazione ospitata dal sistema a Parigi tenta di inserire i messaggi in INVENTQ, PARIS definisce automaticamente un canale mittente del cluster per connettersi al canale ricevente del cluster INVENTORY . NEWYORK. L'applicazione può ricevere risposte quando il nome del gestore code è specificato come gestore code di destinazione e viene fornita una coda di risposta.

Aggiunta di un gestore code a un cluster: code di trasmissione separate

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Prima di iniziare

- Il gestore code non è un membro di alcun cluster.
- Il cluster esiste; esiste un repository completo a cui questo gestore code può connettersi direttamente e il repository è disponibile. Per la procedura per creare il cluster, consultare [“Configurazione di un nuovo cluster”](#) a pagina 318.

Informazioni su questa attività

Questa attività è un'alternativa a [“Aggiunta di un gestore code a un cluster”](#) a pagina 329, in cui si aggiunge un gestore code a un cluster che posiziona i messaggi cluster su una singola coda di trasmissione.

In questa attività, si aggiunge un gestore code a un cluster che crea automaticamente code di trasmissione cluster separate per ogni canale mittente del cluster.

Per mantenere il numero di definizioni di code piccole, il valore predefinito è di utilizzare una singola coda di trasmissione. L'utilizzo di code di trasmissione separate è vantaggioso se si desidera monitorare il traffico destinato a gestori code e cluster differenti. Si potrebbe anche voler separare il traffico verso destinazioni differenti per raggiungere gli obiettivi di isolamento o di prestazioni.

Procedura

1. Modificare il tipo di coda di trasmissione del canale cluster predefinito.

Modificare il gestore code PARIS:

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Ogni volta che il gestore code crea un canale mittente del cluster per inviare un messaggio a un gestore code, crea una coda di trasmissione del cluster. La coda di trasmissione viene utilizzata solo da questo canale mittente del cluster. La coda di trasmissione è permanente - dinamica. Viene creato dalla coda modello, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE, con il nome SYSTEM.CLUSTER.TRANSMIT.ChannelName.



Attenzione: Se si stanno utilizzando delle SYSTEM.CLUSTER.TRANSMIT.QUEUES dedicate con un gestore code che era stato aggiornato da una versione del prodotto antecedente a IBM WebSphere MQ 7.5, assicurarsi che laSYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE abbia l'opzione SHARE/NOSHARE impostata su **SHARE**.

2. Decidere quale repository completo PARIS fa riferimento per primo.

Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi. Raccoglie le informazioni sul cluster da un repository completo e quindi crea il proprio repository parziale. Scegliere uno dei repository come repository completo. Non appena un nuovo gestore code viene aggiunto al cluster, viene immediatamente a conoscenza dell'altro repository. Le informazioni relative alle modifiche a un gestore code vengono inviate direttamente a due repository. In questo esempio, si collega PARIS al gestore code LONDON, solo per motivi geografici.

Nota: Eseguire i passi rimanenti in qualsiasi ordine, dopo l'avvio del gestore code PARIS .

3. Definire un canale CLUSRCVR sul gestore code PARIS.

Ogni gestore code in un cluster deve definire un canale ricevente del cluster su cui può ricevere i messaggi. Su PARIS, definire:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-receiver channel for queue manager PARIS')
```

Il canale ricevente del cluster annuncia la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster INVENTORY. Non creare definizioni su altri gestori code per l'invio al canale ricevente del cluster INVENTORY.PARIS. Le altre definizioni vengono effettuate automaticamente quando necessario. Vedere [Canali cluster](#).

4. Definire un canale CLUSSDR nel gestore code PARIS.

Quando si aggiunge a un cluster un gestore code che non è un repository completo, si definisce un solo canale mittente del cluster per stabilire una connessione iniziale a tale repository. Fare riferimento a [Canale mittente del cluster: CLUSSDR](#) .

Su PARIS, creare la seguente definizione per un canale CLUSSDR denominato INVENTORY.LONDON nel gestore code con indirizzo di rete LONDON.CHSTORE.COM.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

Il gestore code crea automaticamente la coda di trasmissione del cluster dinamico permanente SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON dalla coda modello SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. Imposta l'attributo CLCHNAME della coda di trasmissione su INVENTORY.LONDON.

Risultati

La seguente figura mostra il cluster configurato da questa attività.

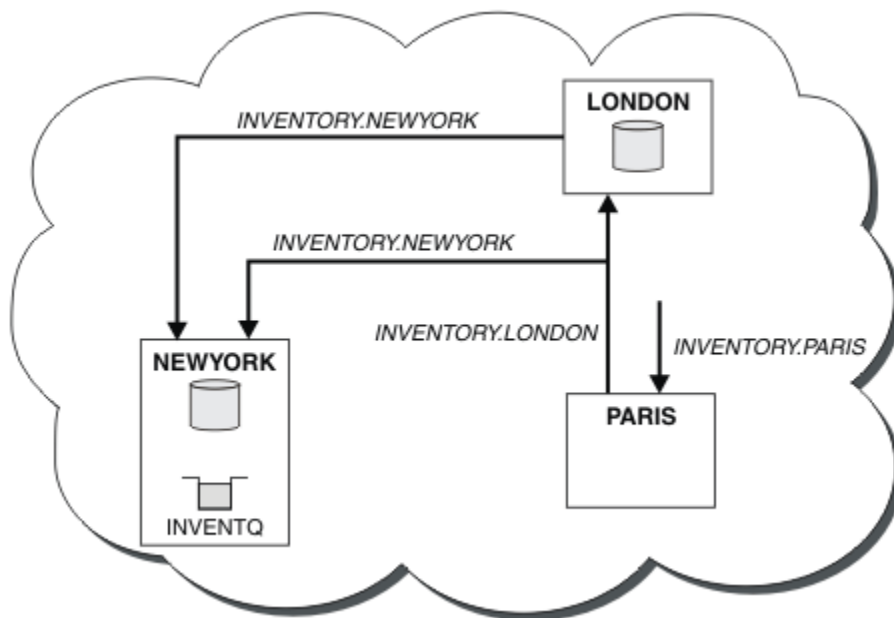


Figura 40. Il cluster INVENTORY con tre gestori code

Creando solo due definizioni, una CLUSRCVR e una CLUSSDR , è stato aggiunto il gestore code PARIS al cluster.

Ora il gestore code PARIS apprende, dal repository completo in LONDON, che la coda INVENTQ è ospitata dal gestore code NEWYORK. Quando un'applicazione ospitata dal sistema a Parigi tenta di inserire i messaggi in INVENTQ, PARIS definisce automaticamente un canale mittente del cluster per connettersi al canale ricevente del cluster INVENTORY . NEWYORK. L'applicazione può ricevere risposte quando il nome del gestore code è specificato come gestore code di destinazione e viene fornita una coda di risposta.

Concetti correlati

Come scegliere quale tipo di coda di trasmissione del cluster utilizzare

Attività correlate

Aggiunta di un gestore code a un cluster utilizzando DHCP

Aggiungere un gestore code a un cluster utilizzando DHCP. L'attività illustra l'omissione del valore CONNAME su una definizione CLUSRCVR .

Aggiunta di un gestore code a un cluster utilizzando DHCP


Aggiungere un gestore code a un cluster utilizzando DHCP. L'attività illustra l'omissione del valore CONNAME su una definizione CLUSRCVR .

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

L'attività dimostra due funzioni speciali:

- La capacità di omettere il valore CONNAME su una definizione CLUSRCVR .
- La capacità di utilizzare +QMNAME+ su una definizione CLUSSDR .

 Nessuna funzione è fornita su z/OS.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in [“Configurazione di un nuovo cluster” a pagina 318](#). Contiene due gestori code, LONDON e NEWYORK, che contengono entrambi repository completi.
- Una nuova filiale della catena di negozi è in fase di configurazione a Parigi e si desidera aggiungere al cluster un gestore code denominato PARIS .
- Il gestore code PARIS invia aggiornamenti di inventario all'applicazione in esecuzione sul sistema a New York inserendo i messaggi nella coda INVENTQ.
- La connettività di rete esiste tra tutti e tre i sistemi.
- Il protocollo di rete è TCP.
- Il sistema del gestore code PARIS utilizza DHCP, il che significa che gli indirizzi IP potrebbero cambiare al riavvio del sistema.
- I canali tra i sistemi PARIS e LONDON vengono denominati in base a una convenzione di denominazione definita. La convenzione utilizza il nome del gestore code del repository completo su LONDON.
- Gli amministratori del gestore code PARIS non hanno informazioni sul nome del gestore code sul repository LONDON . Il nome del gestore code sul repository LONDON è soggetto a modifica.

Informazioni su questa attività

Effettuare le operazioni riportate di seguito per aggiungere un gestore code a un cluster utilizzando DHCP.

Procedura

1. Decidere quale repository completo PARIS fa riferimento per primo.

Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi. Raccoglie le informazioni sul cluster da un repository completo e quindi crea il proprio repository parziale. Scegliere uno dei repository come repository completo. Non appena un nuovo gestore code viene aggiunto al cluster, viene immediatamente a conoscenza dell'altro repository. Le informazioni relative alle modifiche a un gestore code vengono inviate direttamente a due repository. In questo esempio si sceglie di collegare PARIS al gestore code LONDON, solo per motivi geografici.

Nota: Eseguire i passi rimanenti in qualsiasi ordine, dopo l'avvio del gestore code PARIS .

2. Definire un canale CLUSRCVR sul gestore code PARIS.

Ogni gestore code in un cluster deve definire un canale ricevente del cluster su cui può ricevere messaggi. Su PARIS, definire:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

Il canale ricevente del cluster annuncia la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster INVENTORY. Non è necessario specificare CONNAME sul canale ricevente del cluster. È possibile richiedere a IBM MQ di individuare il nome della connessione dal sistema, omettendo CONNAME o specificando CONNAME (' '). IBM MQ genera il valore CONNAME utilizzando l'indirizzo IP corrente del sistema; consultare [CONNAME](#) . Non è necessario creare definizioni su altri gestori code per l'invio al canale ricevente del cluster INVENTORY . PARIS. Le altre definizioni vengono effettuate automaticamente quando necessario.

3. Definire un canale CLUSSDR sul gestore code PARIS.

Ogni gestore code in un cluster deve definire un canale mittente del cluster su cui inviare i messaggi al repository completo iniziale. Su PARIS, creare la seguente definizione per un canale denominato INVENTORY . +QMNAME+ al gestore code con indirizzo di rete LONDON . CHSTORE . COM.

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. Opzionale: Se si sta aggiungendo a un cluster un gestore code che è stato precedentemente rimosso dallo stesso cluster, verificare che venga visualizzato come membro del cluster. In caso contrario, completare i seguenti passi aggiuntivi:

a) Immettere il comando **REFRESH CLUSTER** sul gestore code che si sta aggiungendo.

Questa fase arresta i canali del cluster e fornisce alla tua cache del cluster locale una nuova serie di numeri di sequenza che sono sicuri di essere aggiornati nel resto del cluster.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

Nota: Per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può danneggiare il cluster mentre è in esecuzione e, di nuovo, a intervalli di 27 giorni, quando gli oggetti del cluster inviano automaticamente aggiornamenti sullo stato a tutti i gestori code interessati. Consultare [Refreshing in a large cluster can affect performance and availability of the cluster](#).

b) Riavvia il canale CLUSSDR

(ad esempio, utilizzando il comando `START CHANNEL`).

c) Riavviare il canale CLUSRCVR.

Risultati

Il cluster configurato da questa attività è lo stesso di [“Aggiunta di un gestore code a un cluster”](#) a pagina 329:

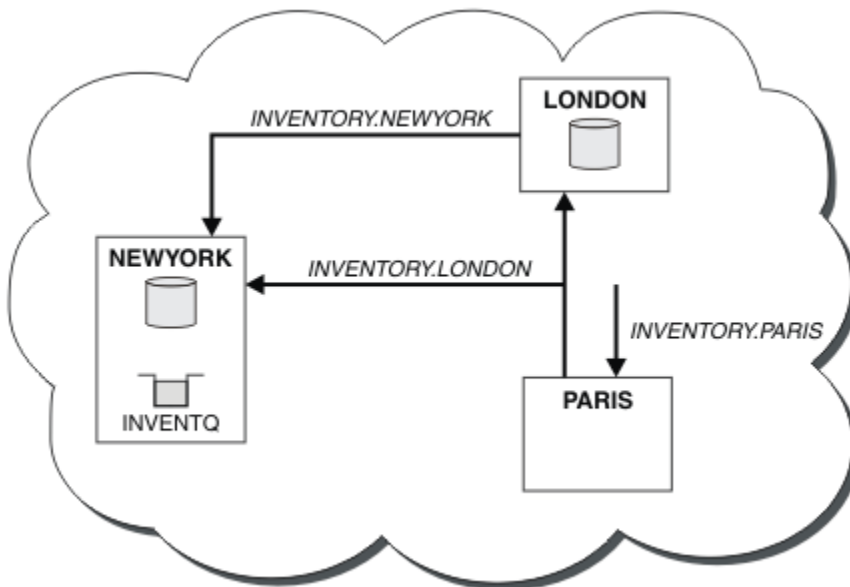


Figura 41. Il cluster INVENTORY con tre gestori code

Creando solo due definizioni, una CLUSRCVR e una definizione CLUSSDR , è stato aggiunto il gestore code PARIS al cluster.

Sul gestore code PARIS , viene avviato il CLUSSDR contenente la stringa +QMNAME+ . Sul sistema LONDON IBM MQ risolve +QMNAME+ nel nome del gestore code (LONDON). IBM MQ mette quindi in corrispondenza la definizione di un canale denominato INVENTORY . LONDON con la corrispondente definizione CLUSRCVR .

IBM MQ restituisce il nome del canale risolto al gestore code PARIS . In PARIS, la definizione di canale CLUSSDR per il canale denominato INVENTORY . +QMNAME+ viene sostituita da una definizione CLUSSDR generata internamente per INVENTORY . LONDON. Questa definizione contiene il nome del canale risolto, ma in caso contrario è uguale alla definizione +QMNAME+ creata. I repository del cluster vengono aggiornati anche con la definizione del canale con il nome del canale appena risolto.

Nota:

1. Il canale creato con il nome +QMNAME+ diventa immediatamente inattivo. Non viene mai utilizzato per trasmettere dati.
2. Le uscite del canale potrebbero vedere la modifica del nome del canale tra una chiamata e la successiva.

Ora il gestore code PARIS apprende, dal repository in LONDON, che la coda INVENTQ è ospitata dal gestore code NEWYORK. Quando un'applicazione ospitata dal sistema a Parigi tenta di inserire i messaggi in INVENTQ, PARIS definisce automaticamente un canale mittente del cluster per connettersi al canale ricevente del cluster INVENTORY . NEWYORK. L'applicazione può ricevere risposte quando il nome del gestore code è specificato come gestore code di destinazione e viene fornita una coda di risposta.

Attività correlate

Aggiunta di un gestore code a un cluster: code di trasmissione separate

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Riferimenti correlati

Definire il canale

Aggiunta di un gestore code su cui è presente una coda

Aggiungere un altro gestore code al cluster, per ospitare un'altra coda INVENTQ . Le richieste vengono inviate alternativamente alle code su ciascun gestore code. Non è necessario apportare modifiche all'host INVENTQ esistente.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in “Aggiunta di un gestore code a un cluster” a pagina 329. Contiene tre gestori code; LONDON e NEWYORK contengono entrambi repository completi, PARIS contiene un repository parziale. L'applicazione di inventario viene eseguita sul sistema a New York, connesso al gestore code NEWYORK . L'applicazione è guidata dall'arrivo di messaggi sulla coda INVENTQ .
- Un nuovo negozio è in fase di allestire a Toronto. Per fornire ulteriore capacità, si desidera eseguire l'applicazione di inventario sul sistema a Toronto e New York.
- La connettività di rete esiste tra tutti e quattro i sistemi.
- Il protocollo di rete è TCP.

Nota: Il gestore code TORONTO contiene solo un repository parziale. Se si desidera aggiungere un gestore code del repository completo a un cluster, fare riferimento a “Spostamento di un repository completo in un altro gestore code” a pagina 340.

Informazioni su questa attività

Effettuare le operazioni riportate di seguito per aggiungere un gestore code su cui è presente una coda.

Procedura

1. Decidere quale repository completo TORONTO fa riferimento per primo.

Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi. Raccoglie le informazioni sul cluster da un repository completo e quindi crea il proprio repository parziale. Non è di particolare importanza quale repository si sceglie. In questo esempio, si sceglie NEWYORK. Una volta che il nuovo gestore code si è unito al cluster, comunica con entrambi i repository.

2. Definire il canale CLUSRCVR .

Ogni gestore code in un cluster deve definire un canale ricevente del cluster su cui può ricevere messaggi. Su TORONTO, definire un canale CLUSRCVR :

```
DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for TORONTO')
```

Il gestore code TORONTO annuncia la propria disponibilità a ricevere messaggi da altri gestori code nel cluster INVENTORY utilizzando il canale ricevente del cluster.

3. Definire un canale CLUSSDR nel gestore code TORONTO.

Ogni gestore code di un cluster deve definire un canale mittente del cluster su cui può inviare messaggi al primo repository completo. In questo caso scegliere NEWYORK. TORONTO necessita della seguente definizione:

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
```

4. Opzionale: Se si sta aggiungendo a un cluster un gestore code che è stato precedentemente rimosso dallo stesso cluster, verificare che venga visualizzato come membro del cluster. In caso contrario, completare i seguenti passi aggiuntivi:

a) Immettere il comando **REFRESH CLUSTER** sul gestore code che si sta aggiungendo.

Questa fase arresta i canali del cluster e fornisce alla tua cache del cluster locale una nuova serie di numeri di sequenza che sono sicuri di essere aggiornati nel resto del cluster.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

Nota: Per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può danneggiare il cluster mentre è in esecuzione e, di nuovo, a intervalli di 27 giorni, quando gli oggetti del cluster inviano automaticamente aggiornamenti sullo stato a tutti i gestori code interessati. Consultare [Refreshing in a large cluster can affect performance and availability of the cluster](#).

b) Riavvia il canale CLUSSDR

(ad esempio, utilizzando il comando [START CHANNEL](#)).

c) Riavviare il canale CLUSRCVR.

5. Esaminare l'applicazione inventario per le affinità dei messaggi.

Prima di procedere, assicurarsi che l'applicazione di inventario non abbia alcuna dipendenza dalla sequenza di elaborazione dei messaggi e installare l'applicazione sul sistema a Toronto.

6. Definire la coda del cluster INVENTQ.

La coda INVENTQ , che è già ospitata dal gestore code NEWYORK , deve essere ospitata anche da TORONTO. Definirlo sul gestore code TORONTO nel modo seguente:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

Risultati

[Figura 42 a pagina 338](#) mostra il cluster INVENTORY configurato da questa attività.

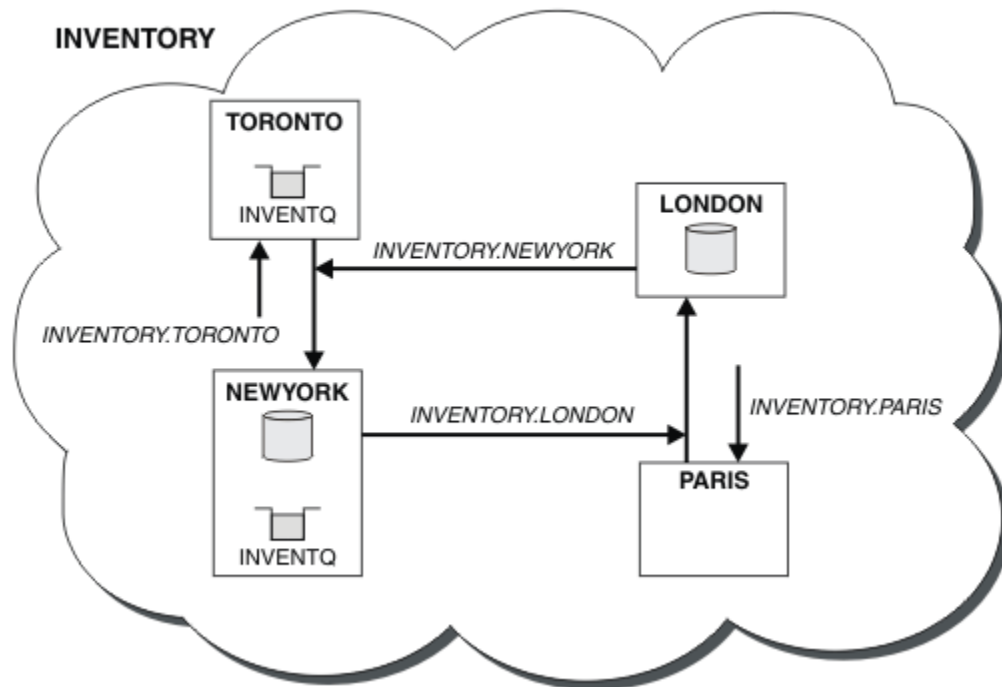


Figura 42. Il cluster INVENTORY con quattro gestori code

La coda INVENTQ e l'applicazione di inventario si trovano ora su due gestori code nel cluster. Ciò aumenta la loro disponibilità, velocizza la velocità di trasmissione dei messaggi e consente la distribuzione del carico di lavoro tra i due gestori code. I messaggi inseriti in INVENTQ da TORONTO o NEWYORK vengono gestiti dall'istanza sul gestore code locale quando possibile. I messaggi immessi da LONDON o PARIS vengono instradati alternativamente a TORONTO o NEWYORK, in modo che il carico di lavoro sia bilanciato.

Questa modifica al cluster è stata effettuata senza che fosse necessario modificare le definizioni sui gestori code NEWYORK, LONDON e PARIS. I repository completi in questi gestori code vengono aggiornati automaticamente con le informazioni necessarie per inviare messaggi a INVENTQ all'indirizzo TORONTO. L'applicazione di inventario continua a funzionare se uno dei gestori code NEWYORK o TORONTO diventa non disponibile e dispone di capacità sufficiente. L'applicazione di inventario deve essere in grado di funzionare correttamente se è ospitata in entrambe le ubicazioni.

Come si può vedere dal risultato di questa attività, è possibile avere la stessa applicazione in esecuzione su più di un gestore code. È possibile eseguire il clustering per distribuire il carico di lavoro in modo uniforme.

Un'applicazione potrebbe non essere in grado di elaborare i record in entrambe le collocazioni. Ad esempio, si supponga di decidere di aggiungere una query di account del cliente e aggiornare l'applicazione in esecuzione in LONDON e NEWYORK. Un record di account può essere conservato solo in un posto. È possibile decidere di controllare la distribuzione delle richieste utilizzando una tecnica di partizionamento dati. È possibile suddividere la distribuzione dei record. È possibile disporre la metà dei record, ad esempio per i numeri di conto 00000 - 49999, da tenere in LONDON. L'altra metà, nell'intervallo 50000 - 99999, è contenuta in NEWYORK. È quindi possibile scrivere un programma di uscita del carico di lavoro del cluster per esaminare il campo account in tutti i messaggi e instradare i messaggi al gestore code appropriato.

Operazioni successive

z/OS Ora che sono state completate tutte le definizioni, se non è stato ancora fatto, avviare l'iniziatore di canali su IBM MQ for z/OS.

Su tutte le piattaforme, avviare un programma listener sul Gestore code TORONTO. Il programma listener attende le richieste di rete in arrivo e avvia il canale ricevente del cluster quando è necessario.

Adding a queue sharing group to existing clusters

Add a queue sharing group on z/OS to existing clusters.

Before you begin

Note:

1. For changes to a cluster to be propagated throughout the cluster, at least one full repository must always be available. Ensure that your repositories are available before starting this task.
2. Queue sharing groups are supported only on IBM MQ for z/OS. This task is not applicable to other platforms.

Scenario:

- The INVENTORY cluster has been set up as described in [“Configurazione di un nuovo cluster”](#) on page 318. It contains two queue managers, LONDON and NEWYORK.
- You want to add a queue sharing group to this cluster. The group, QSGP, comprises three queue managers, P1, P2, and P3. They share an instance of the INVENTQ queue, which is to be defined by P1.

About this task

Follow these steps to add new queue managers that host a shared queue.

Procedure

1. Decide which full repository the queue managers refer to first.

Every queue manager in a cluster must refer to one or other of the full repositories. It gathers information about the cluster from a full repository and so builds up its own partial repository. It is of no particular significance which full repository you choose. In this example, choose NEWYORK. Once the queue sharing group has joined the cluster, it communicates with both of the full repositories.

2. Define the CLUSRCVR channels.

Every queue manager in a cluster needs to define a cluster-receiver channel on which it can receive messages. On P1, P2, and P3, define:

```
DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for sharing queue manager')
```

The cluster-receiver channel advertises the availability of each queue manager to receive messages from other queue managers in the cluster INVENTORY.

3. Define a CLUSSDR channel for the queue sharing group.

Every member of a cluster needs to define one cluster-sender channel on which it can send messages to its first full repository. In this case we have chosen NEWYORK. One of the queue managers in the queue sharing group needs the following group definition. The definition ensures that every queue manager has a cluster-sender channel definition.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
DESCR('Cluster-sender channel to repository at NEWYORK')
```

4. Define the shared queue.

Define the queue INVENTQ on P1 as follows:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

Start the channel initiator and a listener program on the new queue manager. The listener program listens for incoming network requests and starts the cluster-receiver channel when it is needed.

Results

Figure 43 on page 340 shows the cluster set up by this task.

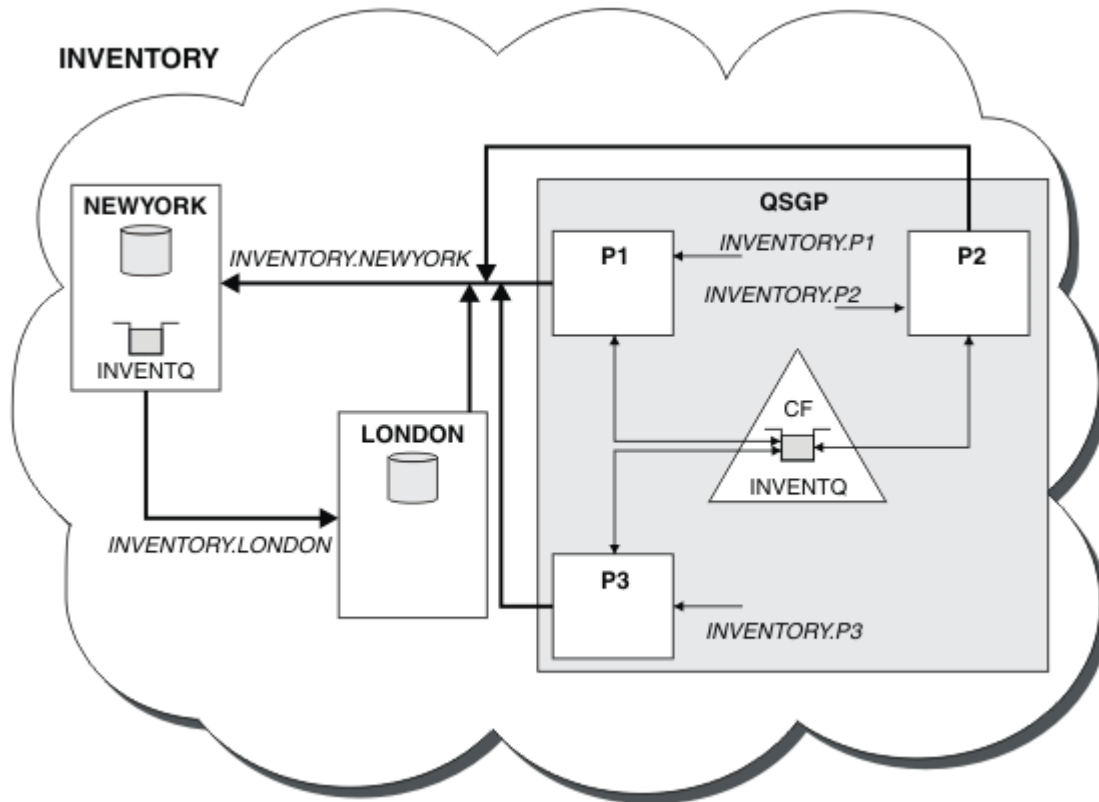


Figure 43. Cluster and queue sharing group

Now messages put on the INVENTQ queue by LONDON are routed alternately around the four queue managers advertised as hosting the queue.

What to do next

A benefit of having members of a queue sharing group host a cluster queue is any member of the group can reply to a request. In this case perhaps P1 becomes unavailable after receiving a message on the shared queue. Another member of the queue sharing group can reply instead.

Spostamento di un repository completo in un altro gestore code

Spostare un repository completo da un gestore code a un altro, creando il nuovo repository dalle informazioni contenute nel secondo repository.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in [“Aggiunta di un gestore code a un cluster”](#) a pagina 329.
- Per motivi aziendali, si desidera ora rimuovere il repository completo dal gestore code LONDON e sostituirlo con un repository completo sul gestore code PARIS. Il gestore code NEWYORK deve continuare a mantenere un repository completo.

Informazioni su questa attività

Attenersi alla seguente procedura per spostare un repository completo in un altro gestore code.

Procedura

1. Modificare PARIS per renderlo un gestore code del repository completo.

Su PARIS, immettere il seguente comando:

```
ALTER QMGR REPOS(INVENTORY)
```

2. Aggiungere un canale CLUSSDR su PARIS

PARIS attualmente ha un canale mittente del cluster che punta a LONDON. LONDON non deve più contenere un repository completo per il cluster. PARIS deve avere un nuovo canale mittente del cluster che punti a NEWYORK, dove ora è conservato l'altro repository completo.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
```

3. Definire un canale CLUSSDR su NEWYORK che punti a PARIS

Attualmente NEWYORK ha un canale mittente del cluster che punta a LONDON. Ora che l'altro repository completo è stato spostato in PARIS, devi aggiungere un nuovo canale mittente del cluster in NEWYORK che punti a PARIS.

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
```

Quando aggiungi il canale mittente del cluster a PARIS, PARIS impara a conoscere il cluster da NEWYORK. Crea il proprio repository completo utilizzando le informazioni da NEWYORK.

4. Verificare che il gestore code PARIS disponga ora di un repository completo

Verificare che il gestore code PARIS abbia creato il proprio repository completo dal repository completo sul gestore code NEWYORK. Immettere i seguenti comandi:

```
DIS QCLUSTER(*) CLUSTER (INVENTORY)  
DIS CLUSQMGR(*) CLUSTER (INVENTORY)
```

Verificare che questi comandi mostrino i dettagli delle stesse risorse in questo cluster come su NEWYORK.

Nota: Se il gestore code NEWYORK non è disponibile, non è possibile completare questa creazione di informazioni. Non passare al passo successivo fino a quando l'attività non è completa.

5. Modificare la definizione del gestore code su LONDON

Infine, modificare il gestore code in LONDON in modo che non contenga più un repository completo per il cluster. Su LONDON, immettere il comando:

```
ALTER QMGR REPOS(' ')
```

Il gestore code non riceve più le informazioni sul cluster. Dopo 30 giorni le informazioni memorizzate nel relativo repository completo scadono. Il gestore code LONDON ora crea il proprio repository parziale.

6. Eliminare o modificare le definizioni in sospeso.

Quando si è certi che la nuova disposizione del cluster funziona come previsto, rimuovere o modificare manualmente le definizioni CLUSSDR definite che non sono più corrette.

- Sul gestore code PARIS , è necessario arrestare ed eliminare il canale mittente del cluster su LONDON, quindi immettere il comando di avvio del canale in modo che il cluster possa utilizzare nuovamente i canali automatici:

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
START CHANNEL(INVENTORY.LONDON)
```

- Sul gestore code NEWYORK , è necessario arrestare ed eliminare il canale mittente del cluster su LONDON, quindi immettere il comando di avvio del canale in modo che il cluster possa utilizzare nuovamente i canali automatici:

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
START CHANNEL(INVENTORY.LONDON)
```

- Sostituire tutti gli altri canali mittenti del cluster definiti manualmente che puntano a LONDON su tutti i gestori code nel cluster con canali che puntano a NEWYORK o PARIS. Dopo aver eliminato un canale, immettere sempre il comando **start channel** in modo che il cluster possa utilizzare nuovamente i canali automatici. In questo piccolo esempio, non ce ne sono altri. Per verificare la presenza di altri elementi dimenticati, immettere il comando `DISPLAY CHANNEL` da ogni gestore code, specificando `TYPE(CLUSSDR)`. Ad esempio:

```
DISPLAY CHANNEL(*) TYPE(CLUSSDR)
```

È importante eseguire questa attività il più presto possibile dopo aver spostato il repository completo da LONDON a PARIS. Prima di eseguire questa attività, i gestori code che hanno definito manualmente i canali CLUSSDR denominati `INVENTORY.LONDON` potrebbero inviare richieste di informazioni utilizzando questo canale.

Dopo che LONDON ha cessato di essere un repository completo, se riceve tali richieste scriverà messaggi di errore nel log degli errori del gestore code. I seguenti esempi mostrano quali messaggi di errore potrebbero essere visualizzati su LONDON:

- AMQ9428: Unexpected publication of a cluster queue object received
- AMQ9432: Query received by a non-repository queue manager

Il gestore code LONDON non risponde alle richieste di informazioni perché non è più un repository completo. I gestori code che richiedono le informazioni da LONDON devono basarsi su NEWYORK per le informazioni sul cluster fino a quando le loro definizioni CLUSSDR definite manualmente non vengono corrette per puntare a PARIS. Questa situazione non deve essere tollerata come una configurazione valida a lungo termine.

Risultati

[Figura 44 a pagina 343](#) mostra il cluster impostato da questa attività.

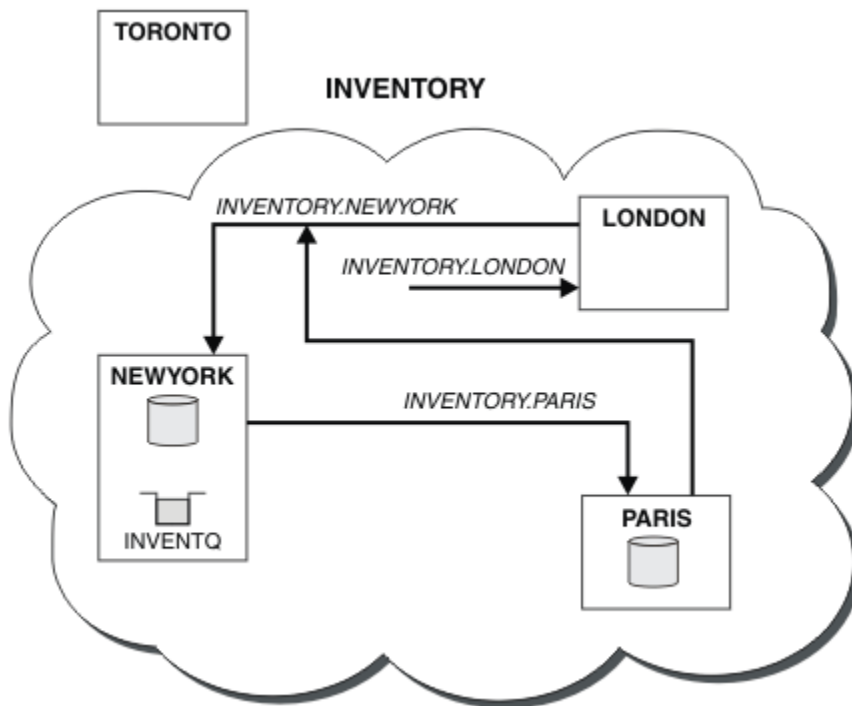


Figura 44. Il cluster INVENTORY con il repository completo è stato spostato in PARIS

Conversione di una rete esistente in un cluster

Convertire una rete di accodamento distribuita esistente in un cluster e aggiungere un ulteriore gestore code per incrementare la capacità.

Prima di iniziare

In “Configurazione di un nuovo cluster” a pagina 318 tramite “Spostamento di un repository completo in un altro gestore code” a pagina 340 hai creato ed esteso un nuovo cluster. Le due attività successive esplorano un approccio diverso: quello di convertire una rete esistente di gestori code in un cluster.

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Una rete IBM MQ è già attiva, collegando le filiali nazionali di una catena di negozi. Ha una struttura hub e spoke: tutti i gestori code sono connessi a un unico gestore code centrale. Il gestore code centrale si trova sul sistema su cui viene eseguita l'applicazione di inventario. L'applicazione è guidata dall'arrivo dei messaggi sulla coda INVENTQ, per cui ciascun gestore code ha una definizione di coda remota.

Questa rete è illustrata in [Figura 45 a pagina 344](#).

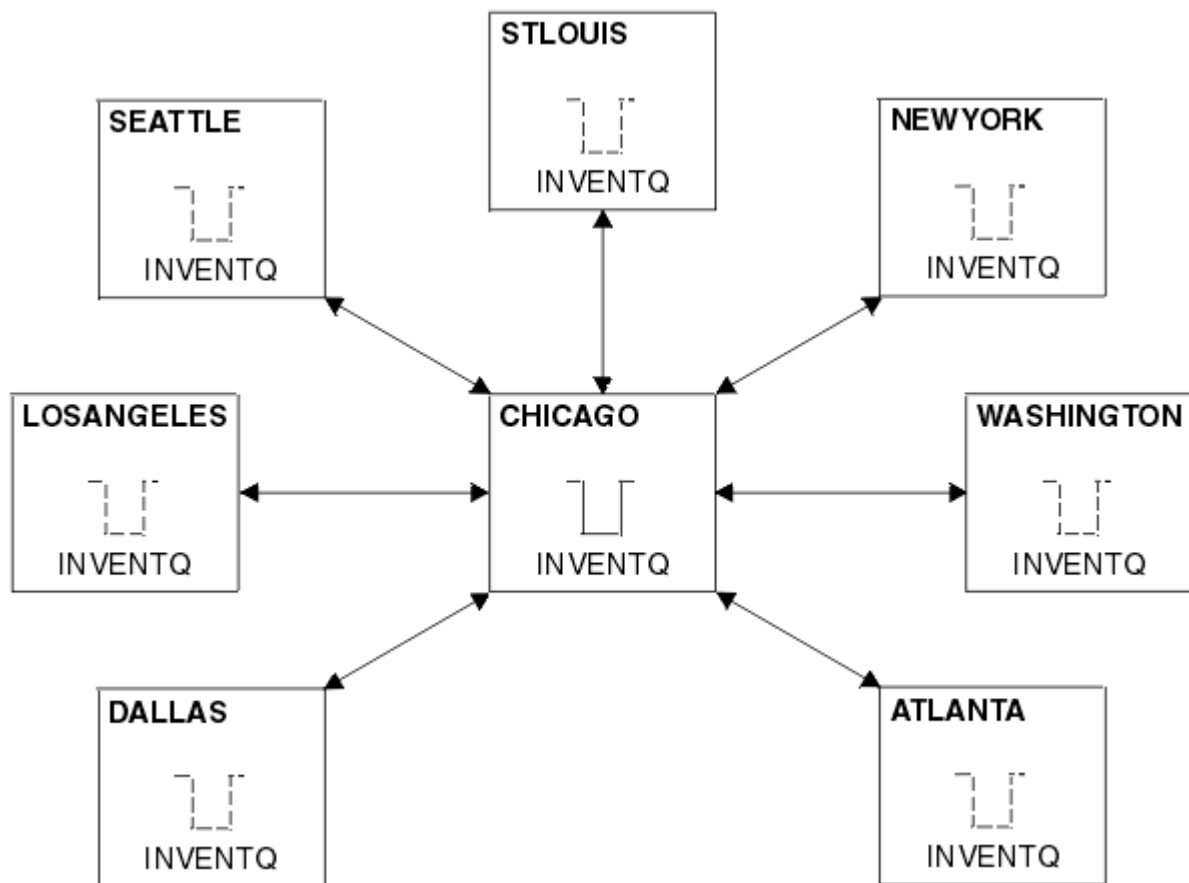


Figura 45. Una rete hub e spoke

- Per facilitare la gestione, si sta per convertire questa rete in un cluster e creare un altro gestore code sul sito centrale per condividere il carico di lavoro.

Il nome cluster è CHNSTORE.

Nota: Il nome cluster CHNSTORE è stato selezionato per consentire la creazione dei nomi dei canali riceventi del cluster utilizzando i nomi nel formato `cluster_name.queue_manager_name` che non superano la lunghezza massima di 20 caratteri, ad esempio CHNSTORE.WASHINGTON.

- Entrambi i gestori code centrali devono ospitare repository completi e devono essere accessibili all'applicazione di inventario.
- L'applicazione di inventario deve essere guidata dall'arrivo di messaggi sulla coda INVENTQ ospitata da uno dei gestori code centrali.
- L'applicazione di inventario deve essere l'unica applicazione in esecuzione in parallelo e accessibile da più di un gestore code. Tutte le altre applicazioni continuano ad essere eseguite come prima.
- Tutte le filiali hanno la connettività di rete ai due gestori code centrali.
- Il protocollo di rete è TCP.

Informazioni su questa attività

Seguire questa procedura per convertire una rete esistente in un cluster.

Procedura

1. Esaminare l'applicazione inventario per le affinità dei messaggi.

Prima di procedere, verificare che l'applicazione possa gestire le affinità dei messaggi. Le affinità di messaggi sono le relazioni tra i messaggi di conversazione scambiati tra due applicazioni, dove i messaggi devono essere elaborati da un particolare gestore code o in una particolare sequenza. Per

ulteriori informazioni sulle affinità dei messaggi, consultare: [“Gestione delle affinità dei messaggi” a pagina 423](#)

2. Modificare i due gestori code centrali per renderli gestori code con repository completo.

I due gestori code CHICAGO e CHICAG02 si trovano nell'hub di questa rete. Si è deciso di concentrare tutte le attività associate al cluster di negozi della catena su questi due gestori code. Oltre all'applicazione di inventario e le definizioni per la coda INVENTQ, si desidera che questi gestori code ospitino i due repository completi per il cluster. Su ciascuno dei due gestori code, immettere il seguente comando:

```
ALTER QMGR REPOS(CHNSTORE)
```

3. Definire un canale CLUSRCVR su ciascun gestore code.

In ogni gestore code del cluster, definire un canale ricevente del cluster e un canale mittente del cluster. Non importa quale canale si definisce per primo.

Creare una definizione CLUSRCVR per pubblicizzare ogni gestore code, il relativo indirizzo di rete e altre informazioni sul cluster. Ad esempio, sul gestore code ATLANTA:

```
DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)  
DESCR('Cluster-receiver channel')
```

4. Definire un canale CLUSSDR su ciascun gestore code

Creare una definizione CLUSSDR su ogni gestore code per collegare tale gestore code a uno o più gestori code del repository completo. Ad esempio, è possibile collegare ATLANTA a CHICAG02:

```
DEFINE CHANNEL(CHNSTORE.CHICAG02) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(CHICAG02.CHSTORE.COM) CLUSTER(CHNSTORE)  
DESCR('Cluster-sender channel to repository queue manager')
```

5. Installare l'applicazione di inventario su CHICAG02.

Si dispone già dell'applicazione di inventario sul gestore code CHICAGO. Ora è necessario eseguire una copia di questa applicazione sul gestore code CHICAG02.


6. Definire la coda INVENTQ sui gestori code centrali.

Su CHICAGO, modificare la definizione della coda locale per la coda INVENTQ per renderla disponibile per il cluster. Immettere il seguente comando:

```
ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

In CHICAG02, creare una definizione per la stessa coda:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

 Su z/OS, è possibile utilizzare l'opzione MAKEDEF della funzione COMMAND di **CSQUTIL** per eseguire una copia esatta su CHICAG02 di INVENTQ su CHICAGO.

Quando si effettuano queste definizioni, viene inviato un messaggio ai repository completi in CHICAGO e CHICAG02 e le informazioni in essi contenute vengono aggiornate. Il gestore code rileva dai repository completi quando inserisce un messaggio in INVENTQ, che esiste una scelta di destinazioni per i messaggi.

7. Verificare che le modifiche al cluster siano state propagate.

Verificare che le definizioni create nel passo precedente siano state propagate attraverso il cluster. Immettere il seguente comando su un gestore code del repository completo:

Aggiunta di un nuovo cluster interconnesso

Aggiungere un nuovo cluster che condivide alcuni gestori code con un cluster esistente.

Prima di iniziare

Nota:

1. Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.
2. Prima di avviare questa attività, verificare la presenza di conflitti di nomi coda e comprendere le conseguenze. Potrebbe essere necessario ridenominare una coda o impostare gli alias della coda prima di continuare.

Scenario:

- Un cluster IBM MQ è stato impostato come descritto in [“Conversione di una rete esistente in un cluster”](#) a pagina 343.
- Deve essere implementato un nuovo cluster denominato MAILORDER . Questo cluster comprende quattro dei gestori code presenti nel cluster CHNSTORE ; CHICAGO, CHICAGO2, SEATTLEe ATLANTAe due ulteriori gestori code; HARTFORD e OMAHA. L'applicazione MAILORDER viene eseguita sul sistema in Omaha, connesso al gestore code OMAHA. Viene gestito dagli altri gestori code nel cluster che inserendo i messaggi sulla coda MORDERQ .
- I repository completi per il cluster MAILORDER sono conservati sui due gestori code CHICAGO e CHICAGO2.
- Il protocollo di rete è TCP.

Informazioni su questa attività

Seguire questi passi per aggiungere un nuovo cluster interconnesso.

Procedura

1. Creare un elenco nomi dei nomi cluster.

I gestori code del repository completo in CHICAGO e in CHICAGO2 ora conterranno i repository completi per entrambi i cluster CHNSTORE e MAILORDER. Innanzitutto, creare un elenco nomi contenente i nomi dei cluster. Definire l'elenco nomi su CHICAGO e CHICAGO2, come segue:

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

2. Modificare le definizioni dei gestori code.

Modificare ora le due definizioni del gestore code in CHICAGO e CHICAGO2. Attualmente queste definizioni mostrano che i gestori code contengono repository completi per il cluster CHNSTORE. Modificare tale definizione per mostrare che i gestori code contengono repository completi per tutti i cluster elencati nell'elenco nomi CHAINMAIL . Modificare le definizioni dei gestori code CHICAGO e CHICAGO2 :

```
ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
```

3. Modificare i canali CLUSRCVR su CHICAGO e CHICAGO2.

Le definizioni di canale CLUSRCVR in CHICAGO e CHICAGO2 mostrano che i canali sono disponibili nel cluster CHNSTORE. È necessario modificare la definizione del ricevente del cluster per mostrare che i canali sono disponibili per tutti i cluster elencati nell'elenco nomi CHAINMAIL . Modificare la definizione del ricevente cluster in CHICAGO:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

In CHICAGO2, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

4. Modificare i canali CLUSSDR su CHICAGO e CHICAGO2.

Modificare le definizioni di canale CLUSSDR per aggiungere l'elenco nomi. In CHICAGO, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

In CHICAGO2, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

5. Creare un elenco nomi su SEATTLE e ATLANTA.

Poiché SEATTLE e ATLANTA saranno membri di più di un cluster, è necessario creare un elenco nomi contenente i nomi dei cluster. Definire l'elenco nomi su SEATTLE e ATLANTA, come segue:

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

6. Modificare i canali CLUSRCVR su SEATTLE e ATLANTA.

Le definizioni di canale CLUSRCVR in SEATTLE e ATLANTA mostrano che i canali sono disponibili nel cluster CHNSTORE. Modificare le definizioni di canali di ricezione cluster per mostrare che i canali sono disponibili per tutti i cluster elencati nell'elenco nomi CHAINMAIL . In SEATTLE, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

In ATLANTA, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

7. Modificare i canali CLUSSDR su SEATTLE e ATLANTA.

Modificare le definizioni di canale CLUSSDR per aggiungere l'elenco nomi. In SEATTLE, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

In ATLANTA, immettere il comando:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

8. Definire i canali CLUSRCVR e CLUSSDR su HARTFORD e OMAHA.

Nei due nuovi gestori code HARTFORD e OMAHA, definire i canali ricevente e mittente del cluster. Non importa in quale sequenza si fanno le definizioni. In HARTFORD, immettere:

```
DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
```

In OMAHA, immettere:

```
DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
```

9. Definire la coda MORDERQ su OMAHA.

Il passo finale per completare questa attività consiste nel definire la coda MORDERQ sul gestore code OMAHA. In OMAHA, immettere:

```
DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
```

10. Verificare che le modifiche al cluster siano state propagate.

Verificare che le definizioni create con le operazioni precedenti siano state propagate attraverso il cluster. Immettere i comandi riportati di seguito su un gestore code del repository completo:

```
DIS QCLUSTER (MORDERQ)
DIS CLUSQMGR
```

11.

Risultati

Il cluster configurato da questa attività è mostrato in [Figura 46 a pagina 349](#).

Ora abbiamo due cluster che si sovrappongono. I repository completi per entrambi i cluster si trovano in CHICAGO e CHICAGO2. L'applicazione dell'ordine di posta in esecuzione su OMAHA è indipendente dall'applicazione dell'inventario in esecuzione su CHICAGO. Tuttavia, alcuni dei gestori code che si trovano nel cluster CHNSTORE si trovano anche nel cluster MAILORDER e possono quindi inviare messaggi a entrambe le applicazioni. Prima di eseguire questa attività per sovrapporre due cluster, tenere presente la possibilità di conflitti tra nomi di coda.

Si supponga che su NEWYORK nel cluster CHNSTORE e su OMAHA in cluster MAILORDER, vi sia una coda denominata ACCOUNTQ. Se si sovrappongono i cluster e quindi un'applicazione su SEATTLE inserisce un messaggio nella coda ACCOUNTQ, il messaggio può andare a una delle istanze di ACCOUNTQ.

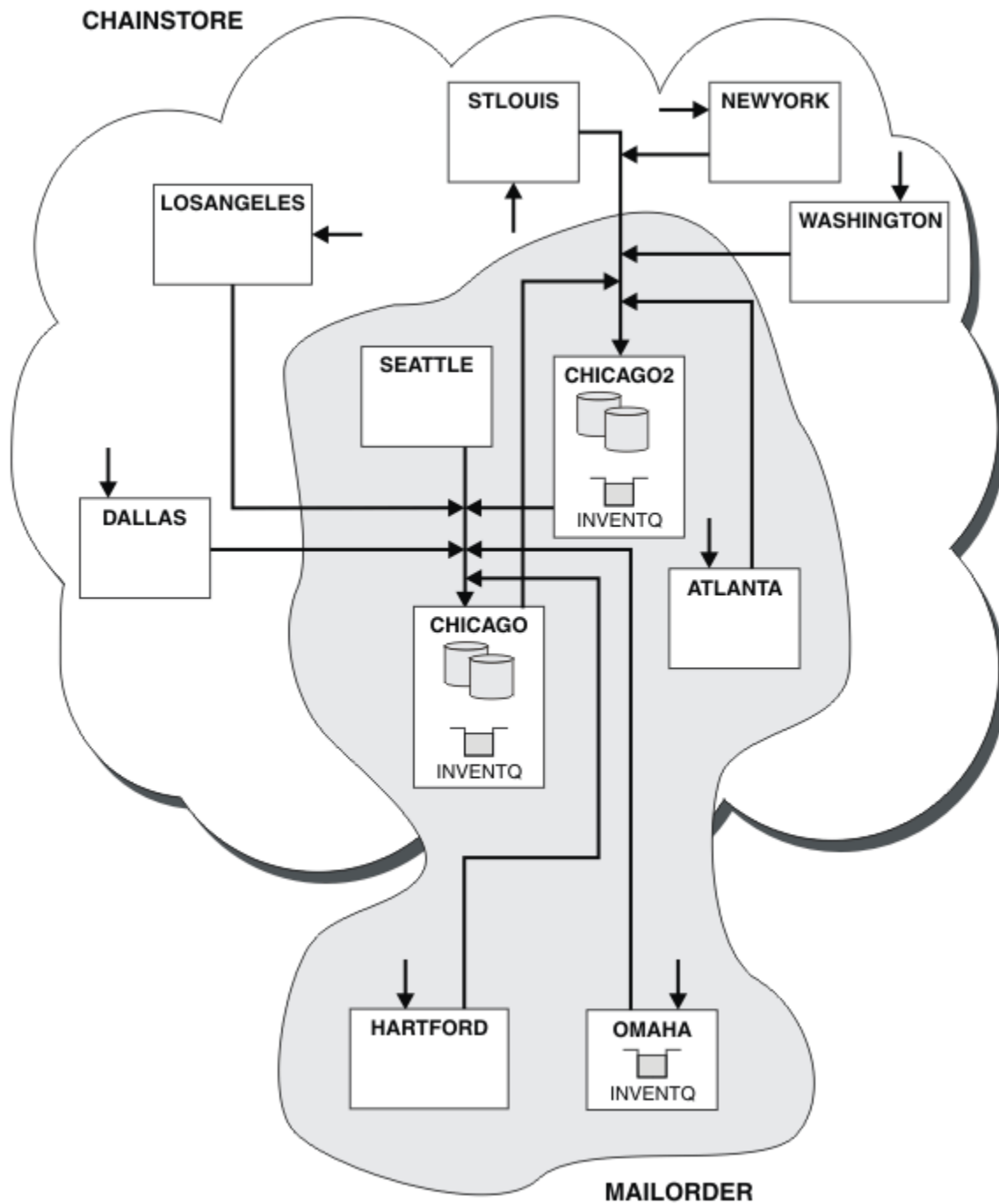


Figura 46. Cluster interconnessi

Operazioni successive

Si supponga di decidere di unire il cluster MAILORDER con quello CHNSTORE per formare un cluster di grandi dimensioni denominato CHNSTORE.

Per unire il cluster MAILORDER con il cluster CHNSTORE, in modo che CHICAGO e CHICAGO2 conservino i repository completi:

- Modificare le definizioni del gestore code per CHICAGO e CHICAG02, rimuovendo l'attributo REPOSNL, che specifica l'elenco nomi (CHAINMAIL) e sostituendolo con un attributo REPOS che specifica il nome cluster (CHNSTORE). Ad esempio:

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- Su ciascun gestore code nel cluster MAILORDER, modificare tutte le definizioni di canale e di coda per modificare il valore dell'attributo CLUSTER da MAILORDER a CHNSTORE. Ad esempio, in HARTFORD, immettere:

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

In OMAHA immettere:

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- Modificare tutte le definizioni che specificano l'elenco nomi cluster CHAINMAIL, ossia le definizioni del canale CLUSRCVR e CLUSSDR in CHICAGO, CHICAG02, SEATTLEe ATLANTA, per indicare invece il cluster CHNSTORE.

Da questo esempio, è possibile vedere il vantaggio di utilizzare gli elenchi nomi. Invece di modificare le definizioni del gestore code per CHICAGO e CHICAG02, è possibile modificare il valore dell'elenco nomi CHAINMAIL. Allo stesso modo, invece di modificare le definizioni di canale CLUSRCVR e CLUSSDR all'indirizzo CHICAGO, CHICAG02, SEATTLEe ATLANTA, è possibile ottenere il risultato richiesto modificando l'elenco nomi.

Attività correlate

Rimozione di una rete cluster

Rimuovere un cluster da una rete e ripristinare la configurazione dell'accodamento distribuito.

Rimozione di una rete cluster

Rimuovere un cluster da una rete e ripristinare la configurazione dell'accodamento distribuito.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Un cluster IBM MQ è stato configurato come descritto in [“Conversione di una rete esistente in un cluster”](#) a pagina 343.
- Questo cluster deve essere rimosso dal sistema. La rete di gestori code deve continuare a funzionare come prima dell'implementazione del cluster.

Informazioni su questa attività

Seguire questa procedura per rimuovere una rete cluster.

Procedura

1. Rimuovere le code cluster dal cluster CHNSTORE.

Su CHICAGO e CHICAG02, modificare la definizione della coda locale per la coda INVENTQ per eliminare la coda dal cluster. Immettere il seguente comando:

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

Quando si modifica la coda, le informazioni nei repository completi vengono aggiornate e propagate in tutto il cluster. Le applicazioni attive che utilizzano MQ00_BIND_NOT_FIXED e le applicazioni che utilizzano MQ00_BIND_AS_Q_DEF in cui la coda è stata definita con DEFBIND(NOTFIXED), non riescono alla successiva chiamata MQPUT o MQPUT1 tentata. Viene restituito il codice di errore MQRC_UNKNOWN_OBJECT_NAME.

Non è necessario eseguire prima il passo 1, ma in caso contrario, eseguirlo dopo il passo 4.

2. Arrestare tutte le applicazioni che hanno accesso alla coda cluster.

Arrestare tutte le applicazioni che hanno accesso alle code cluster. In caso contrario, alcune informazioni sul cluster potrebbero rimanere sul gestore code locale quando si aggiorna il cluster nel passo 5. Queste informazioni vengono rimosse quando tutte le applicazioni sono state arrestate e i canali cluster sono stati disconnessi.

3. Rimuovere l'attributo repository dai gestori code del repository completo.

Su CHICAGO e CHICAGO2, modificare le definizioni del gestore code in modo da rimuovere l'attributo del repository. Per eseguire questa operazione, immettere il comando:

```
ALTER QMGR REPOS(' ')
```

I gestori code informano gli altri gestori code nel cluster che non detengono più i repository completi. Quando gli altri gestori code ricevono queste informazioni, viene visualizzato un messaggio che indica che il repository completo è terminato. Vengono inoltre visualizzati uno o più messaggi che indicano che non sono più disponibili repository per il cluster CHNSTORE.

4. Rimuovere i canali cluster.

Su CHICAGO, eliminare i canali cluster:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

Nota: È importante immettere prima il comando CLUSSDR, quindi il comando CLUSRCVR. Non immettere prima il comando CLUSRCVR, quindi il comando CLUSSDR. In questo modo, vengono creati canali in dubbio con stato ARRESTATO. È quindi necessario emettere un comando START CHANNEL per ripristinare i canali arrestati; ad esempio, START CHANNEL(CHNSTORE.CHICAGO).

Vengono visualizzati messaggi che indicano che non vi sono repository per il cluster CHNSTORE.

Se non sono state rimosse le code del cluster come descritto nel passo 1, procedere ora.

5. Arrestare i canali cluster.

Su CHICAGO arrestare i canali cluster con i seguenti comandi:

```
STOP CHANNEL(CHNSTORE.CHICAGO2)
STOP CHANNEL(CHNSTORE.CHICAGO)
```

6. Ripetere i passi 4 e 5 per ogni gestore code del cluster.
7. Arrestare i canali cluster, quindi rimuovere tutte le definizioni per i canali cluster e le code cluster da ogni gestore code.
8. Opzionale: Cancellare le informazioni sul cluster memorizzato nella cache contenute nel gestore code.

Anche se i gestori code non sono più membri del cluster, ciascuno di essi conserva una copia memorizzata nella cache delle informazioni sul cluster. Se si desidera rimuovere questi dati, consultare l'attività [“Ripristino di un gestore code allo stato pre - cluster”](#) a pagina 379.

9. Sostituire le definizioni della coda remota per INVENTQ

In modo che la rete possa continuare a funzionare, sostituire la definizione della coda remota per INVENTQ in ogni gestore code.

10. Riordinare il cluster.

Eliminare tutte le definizioni di coda o canale non più necessarie.

Attività correlate

Aggiunta di un nuovo cluster interconnesso

Aggiungere un nuovo cluster che condivide alcuni gestori code con un cluster esistente.

Creazione di due cluster sovrapposti con un gestore code del gateway

Seguire le istruzioni nell'attività per creare cluster sovrapposti con un gestore code del gateway. Utilizzare i cluster come punto iniziale per i seguenti esempi di isolamento dei messaggi in un'applicazione da messaggi in altre applicazioni in un cluster.

Informazioni su questa attività

La configurazione cluster di esempio utilizzata per illustrare l'isolazione del traffico di messaggi cluster viene mostrata in [Figura 47 a pagina 352](#). L'esempio è descritto in [Cluster: isolamento dell'applicazione utilizzando più code di trasmissione cluster](#).

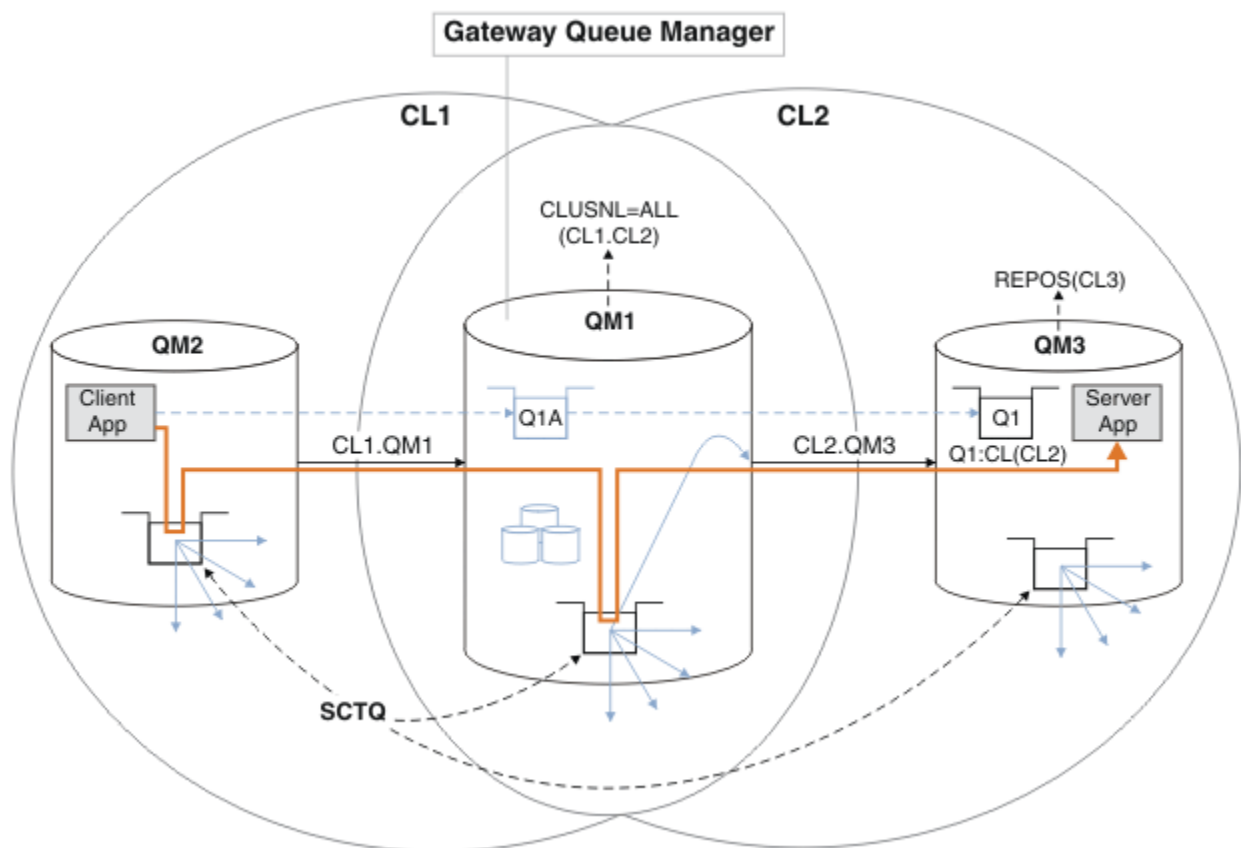


Figura 47. Applicazione client-server distribuita all'architettura hub e spoke utilizzando i cluster IBM MQ

Per rendere il numero di passi per costruire l'esempio il meno possibile, la configurazione è semplice, piuttosto che realistica. L'esempio potrebbe rappresentare l'integrazione di due cluster creati da due organizzazioni separate. Per uno scenario più realistico, consultare [Cluster: Pianificazione della configurazione delle code di trasmissione cluster](#).

Seguire la procedura per costruire i cluster. I cluster vengono utilizzati nei seguenti esempi di isolamento del traffico di messaggi dall'applicazione client all'applicazione server.

Le istruzioni aggiungono un paio di gestori code aggiuntivi in modo che ogni cluster abbia due repository. Il gestore code del gateway non viene utilizzato come repository per motivi di prestazioni.

Procedura

1. Creare e avviare i gestori code QM1, QM2, QM3, QM4, QM5.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QM n
strmqm QmgrName
```

Nota: QM4 e QM5 sono i repository completi di backup per i cluster.

2. Definire e avviare i listener per ciascun gestore code.

```
*... On QM n
DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141 n)
```

3. Creare un elenco di nomi cluster per tutti i cluster.

```
*... On QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. Creare repository completi QM2 e QM4 per CL1, QM3 e QM5 per CL2.

- a) Per CL1:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

- b) Per CL2:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. Aggiungere i canali mittente del cluster e ricevente del cluster per ciascun gestore code e cluster.

Eseguire i seguenti comandi su QM2, QM3, QM4 e QM5, dove *c*, *n* e *m* assumono i valori mostrati in [Tabella 27](#) a pagina 353 per ciascun gestore code:

Gestore code	Cluster <i>c</i>	Altro repository <i>n</i>	Questo repository <i>m</i>
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

```
*... On QM m
DEFINE CHANNEL(CL c.QM n) CHLTYPE(CLUSSDR) CONNAME('localhost(141 n)') CLUSTER(CL c) REPLACE
DEFINE CHANNEL(CL c.QM m) CHLTYPE(CLUSRCVR) CONNAME('localhost(141 m)') CLUSTER(CL c) REPLACE
```

6. Aggiungere il gestore code del gateway, QM1, a ciascuno dei cluster.

```
*... On QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. Aggiungere la coda locale Q1 al gestore code QM3 nel cluster CL2.

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. Aggiungere l'alias del gestore code con cluster Q1A al gestore code del gateway.

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

Nota: Le applicazioni che utilizzano l'alias del gestore code su qualsiasi altro gestore code, ma QM1, devono specificare DEFBIND (NOTFIXED) quando aprono la coda alias. **DEFBIND** specifica se le informazioni di instradamento nell'intestazione del messaggio sono fisse quando la coda viene aperta dall'applicazione. Se è impostato sul valore predefinito, OPEN, i messaggi vengono instradati a Q1@QM1. Q1@QM1 non esiste, quindi i messaggi provenienti da altri gestori code finiscono su una coda di messaggi non recapitabili. Impostando l'attributo della coda su DEFBIND (NOTFIXED), le applicazioni come **amqspout**, che per impostazione predefinita utilizzano l'impostazione della coda di **DEFBIND**, si comportano correttamente.

9. Aggiungere le definizioni di alias del gestore code del cluster per tutti i gestori code del cluster al gestore code del gateway QM1.

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

Suggerimento: Le definizioni di alias del gestore code sul gestore code del gateway trasferiscono i messaggi che fanno riferimento a un gestore code in un altro cluster; consultare [Alias del gestore code cluster](#).

Operazioni successive

1. Verificare la definizione alias della coda inviando un messaggio da QM2 a Q1 su QM3 utilizzando la definizione alias della coda Q1A.
 - a. Eseguire il programma di esempio **amqspout** su QM2 per inserire un messaggio.

```
C:\IBM\MQ>amqspout Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

- b. Eseguire il programma di esempio **amqsget** per richiamare il messaggio da Q1 on QM3

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. Verificare le definizioni alias del gestore code inviando un messaggio di richiesta e ricevendo un messaggio di risposta su una coda di risposta dinamica temporanea.

Il diagramma mostra il percorso utilizzato dal messaggio di risposta per tornare a una coda dinamica temporanea, denominata RQ. L'applicazione server, connessa a QM3, apre la coda di risposta utilizzando il nome del gestore code QM2. Il nome del gestore code QM2 è definito come alias del gestore code in cluster su QM1. QM3 instrada il messaggio di risposta a QM1. QM1 instrada il messaggio a QM2.

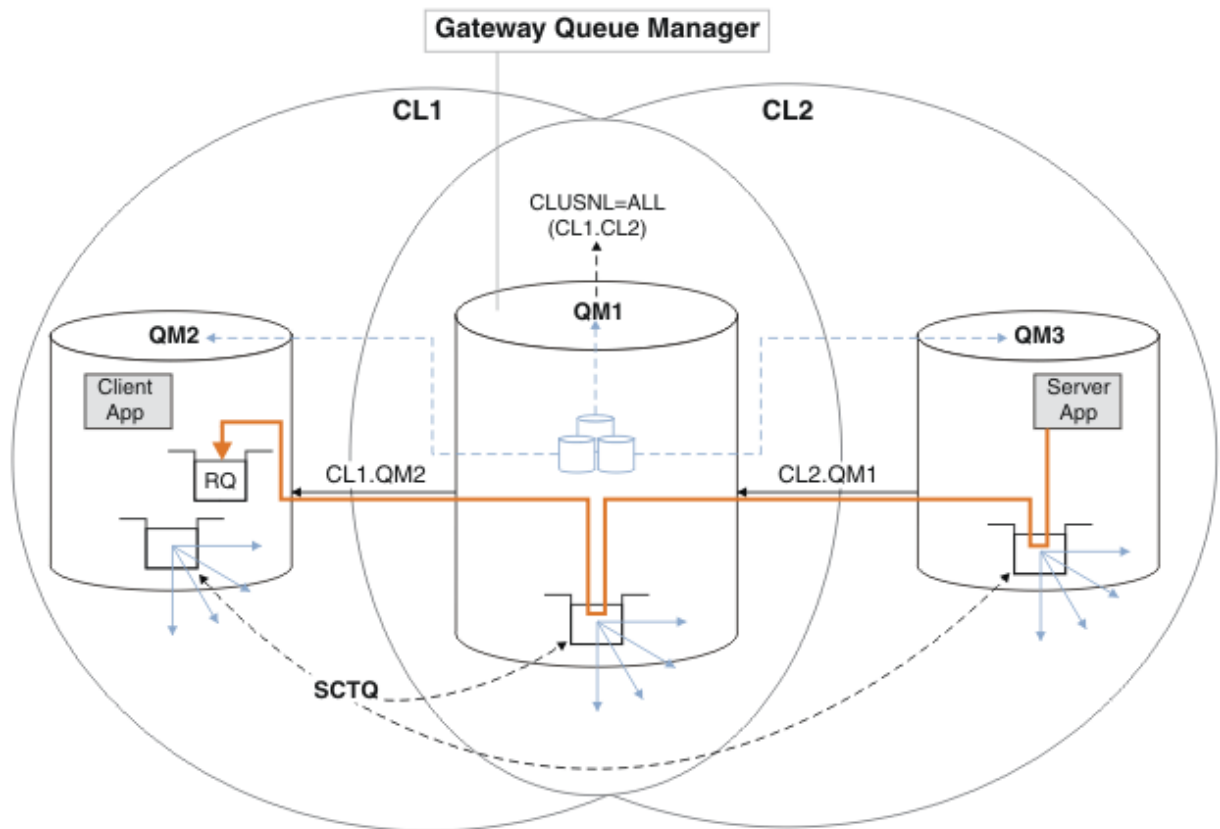


Figura 48. Utilizzo di un alias del gestore code per restituire il messaggio di risposta a un cluster differente

Il modo in cui funziona l'instradamento è il seguente. Ogni gestore code in ogni cluster ha una definizione di alias del gestore code su QM1. Gli alias sono raggruppati in tutti i cluster. Le frecce tratteggiate grigie da ciascuno degli alias a un gestore code mostrano che ogni alias del gestore code viene risolto in un gestore code reale in almeno uno di questi cluster. In questo caso, l'alias QM2 è raggruppatto in cluster CL1 e CL2 e viene risolto nel gestore code reale QM2 in CL1. L'applicazione server crea il messaggio di risposta utilizzando il nome della coda di risposta RQ e il nome del gestore code di risposta QM2. Il messaggio viene instradato a QM1 perché la definizione alias del gestore code QM2 è definita su QM1 nel cluster CL2 e il gestore code QM2 non è nel cluster CL2. Poiché il messaggio non può essere inviato al gestore code di destinazione, viene inviato al gestore code che ha la definizione alias.

QM1 colloca il messaggio nella coda di trasmissione del cluster su QM1 per il trasferimento a QM2. QM1 instrada il messaggio a QM2 perché la definizione dell'alias del gestore code su QM1 per QM2 definisce QM2 come il gestore code di destinazione reale. La definizione non è circolare, poiché le definizioni alias possono fare riferimento solo a definizioni reali; l'alias non può puntare a se stesso. La definizione reale viene risolta da QM1, perché sia QM1 che QM2 si trovano nello stesso cluster, CL1. QM1 rileva le informazioni di collegamento per QM2 dal contenitore per CL1 e instrada il messaggio a QM2. Perché il messaggio venga reinstradato da QM1, l'applicazione server deve aver aperto la coda di risposta con l'opzione DEFBIND impostata su MQBND_BIND_NOT_FIXED. Se l'applicazione server ha aperto la coda di risposta con l'opzione MQBND_BIND_ON_OPEN, il messaggio non viene reinstradato e finisce su una coda di messaggi non recapitabili.

- a. Creare una coda di richieste in cluster con un trigger su QM3.

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

- b. Creare una definizione alias della coda cluster di QR sul gestore code del gateway, QM1.

```
*... On QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

- c. Creare una definizione di processo per avviare il programma echo di esempio **amqsech** su QM3.

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

- d. Creare una coda modello su QM2 per il programma di esempio **amqsreq** per creare la coda di risposta dinamica temporanea.

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

- e. Verificare la definizione alias del gestore code inviando una richiesta da QM2 a QR on QM3 utilizzando la definizione alias della coda QRA.

- i) Eseguire il programma di controllo trigger su QM3.

```
runmqtrm -m QM3
```

L'output è

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
01/02/2012 16:17:15: IBM MQ trigger monitor started.
```

```
-----
01/02/2012 16:17:15: Waiting for a trigger message
```

- ii) Eseguire il programma di esempio **amqsreq** on QM2 per inserire una richiesta e attendere una risposta.

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3

response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

Concetti correlati

Controllo accessi e code di trasmissione di più cluster

Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster

Attività correlate

Clustering: pianificazione della configurazione delle code di trasmissione del cluster

“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare

la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una definizione remota della coda cluster e un canale mittente e una coda di trasmissione separati.

Prima di iniziare

Crea i cluster di sovrapposizione mostrati nell' applicazione client - server distribuita all'architettura hub e spoke utilizzando i cluster IBM MQ in “Creazione di due cluster sovrapposti con un gestore code del gateway” a pagina 352 seguendo i passi in tale attività.

Informazioni su questa attività

La soluzione utilizza l'accodamento distribuito per separare i messaggi per l'applicazione Server App dal traffico di altri messaggi sul gestore code del gateway. È necessario definire una definizione di coda remota con cluster su QM1 per deviare i messaggi su una coda di trasmissione e su un canale diversi. La definizione della coda remota deve includere un riferimento alla specifica coda di trasmissione che memorizza i messaggi solo per Q1 su QM3. In Figura 49 a pagina 357, l'alias della coda cluster Q1A è integrato da una definizione della coda remota Q1Re sono aggiunti una coda di trasmissione e un canale mittente.

In questa soluzione, tutti i messaggi di risposta vengono restituiti utilizzando il comune `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Il vantaggio di questa soluzione è che è facile separare il traffico per più code di destinazione sullo stesso gestore code, nello stesso cluster. Lo svantaggio della soluzione è che non è possibile utilizzare il bilanciamento del carico di lavoro del cluster tra più copie di Q1 su gestori code differenti. Per superare questo svantaggio, consultare “Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 359. È inoltre necessario gestire la commutazione da una coda di trasmissione all'altra.

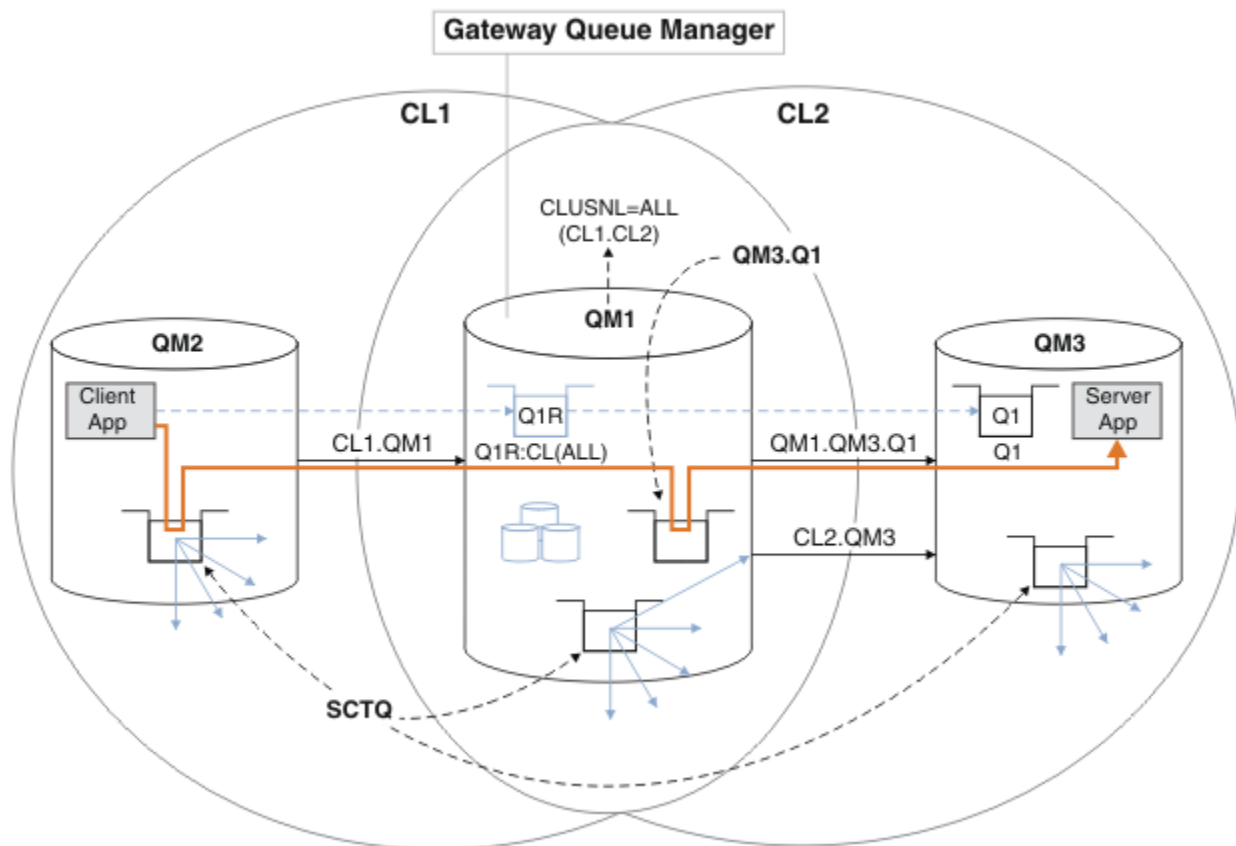


Figura 49. Applicazione client-server distribuita all'architettura del cluster hub e spoke utilizzando definizioni di coda remota

Procedura

1. Creare un canale per separare il traffico di messaggi per Q1 dal gestore code gateway
 - a) Creare un canale mittente sul gestore code del gateway, QM1, sul gestore code di destinazione, QM3.

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
```

- b) Creare un canale ricevente sul gestore code di destinazione, QM3.

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
```

2. Creare una coda di trasmissione sul gestore code del gateway per il traffico di messaggi verso Q1

```
DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE  
START CHANNEL(QM1.QM3.Q1)
```

L'avvio del canale associato alla coda di trasmissione associa la coda di trasmissione al canale. Il canale viene avviato automaticamente, una volta che la coda di trasmissione è stata associata al canale.

3. Integrare la definizione dell'alias della coda con cluster per Q1 sul gestore code del gateway con una definizione della coda remota con cluster.

```
DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNNAME(QM3) XMITQ(QM3.Q1) REPLACE
```

Operazioni successive

Verificare la configurazione inviando un messaggio a Q1 su QM3 da QM2 utilizzando la definizione remota della coda cluster Q1R sul gestore code del gateway QM1.

1. Eseguire il programma di esempio **amqspu**t su QM2 per inserire un messaggio.

```
C:\IBM\MQ>amqspu Q1R QM2  
Sample AMQSPUT0 start  
target queue is Q1R  
Sample request message from QM2 to Q1 using Q1R
```

```
Sample AMQSPUT0 end
```

2. Eseguire il programma di esempio **amqsge**t per richiamare il messaggio da Q1 on QM3

```
C:\IBM\MQ>amqsge Q1 QM3  
Sample AMQSGET0 start  
message <Sample request message from QM2 to Q1 using Q1R>  
no more messages  
Sample AMQSGET0 end
```

Concetti correlati

[Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster](#)

[Controllo accessi e code di trasmissione di più cluster](#)

Attività correlate

[Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway](#)

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una coda

di trasmissione cluster aggiuntiva per separare il traffico di messaggi a un singolo gestore code in un cluster.

Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza un cluster aggiuntivo per isolare i messaggi in una particolare coda cluster.

Modifica del valore predefinito per separare le code di trasmissione del cluster per isolare il traffico dei messaggi

È possibile modificare il modo predefinito in cui un gestore code memorizza i messaggi per una coda cluster o un argomento su una coda di trasmissione. La modifica del valore predefinito fornisce un modo per isolare i messaggi cluster su un gestore code del gateway.

Clustering: pianificazione della configurazione delle code di trasmissione del cluster

“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una coda di trasmissione cluster aggiuntiva per separare il traffico di messaggi a un singolo gestore code in un cluster.

Prima di iniziare

1. Il gestore code del gateway deve essere su IBM MQ.
2. Crea i cluster di sovrapposizione mostrati nell' applicazione client - server distribuita all'architettura hub e spoke utilizzando i cluster IBM MQ in “Creazione di due cluster sovrapposti con un gestore code del gateway” a pagina 352 seguendo i passi in tale attività.

Informazioni su questa attività

Sul gestore code del gateway, QM1, aggiungere una coda di trasmissione e impostarne l'attributo di coda CLCHNAME. Impostare CLCHNAME sul nome del canale ricevente del cluster su QM3 ; consultare Figura 50 a pagina 360.

Questa soluzione presenta una serie di vantaggi rispetto alla soluzione descritta in “Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway” a pagina 356:

- Richiede meno definizioni aggiuntive.
- Supporta il bilanciamento del carico di lavoro tra più copie della coda di destinazione, Q1, su gestori code differenti nello stesso cluster, CL2.
- Il gestore code del gateway passa automaticamente alla nuova configurazione quando il canale viene riavviato senza perdere alcun messaggio.
- Il gestore code del gateway continua ad inoltrare i messaggi nello stesso ordine in cui li ha ricevuti. Ciò avviene anche se lo switch si verifica con i messaggi per la coda Q1 a QM3 ancora su SYSTEM.CLUSTER.TRANSMIT.QUEUE.

La configurazione per isolare il traffico di messaggi cluster in Figura 50 a pagina 360 non risulta in un isolamento del traffico così grande come la configurazione che utilizza le code remote in “Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway” a pagina 356. Se il gestore code QM3 in CL2 ospita diverse code cluster e applicazioni server, tutte queste code condividono il canale cluster, CL2.QM3, connettendo QM1 a QM3. I flussi aggiuntivi sono illustrati in

Figura 50 a pagina 360 dalla freccia grigia che rappresenta il potenziale traffico di messaggi cluster da SYSTEM. CLUSTER. TRANSMIT. QUEUE al canale mittente del cluster CL2. QM3.

Il rimedio consiste nel limitare il gestore code ad ospitare una coda cluster in uno specifico cluster. Se il gestore code ospita già un numero di code cluster, per soddisfare questa limitazione, è necessario creare un altro gestore code o un altro cluster; consultare “Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 362.

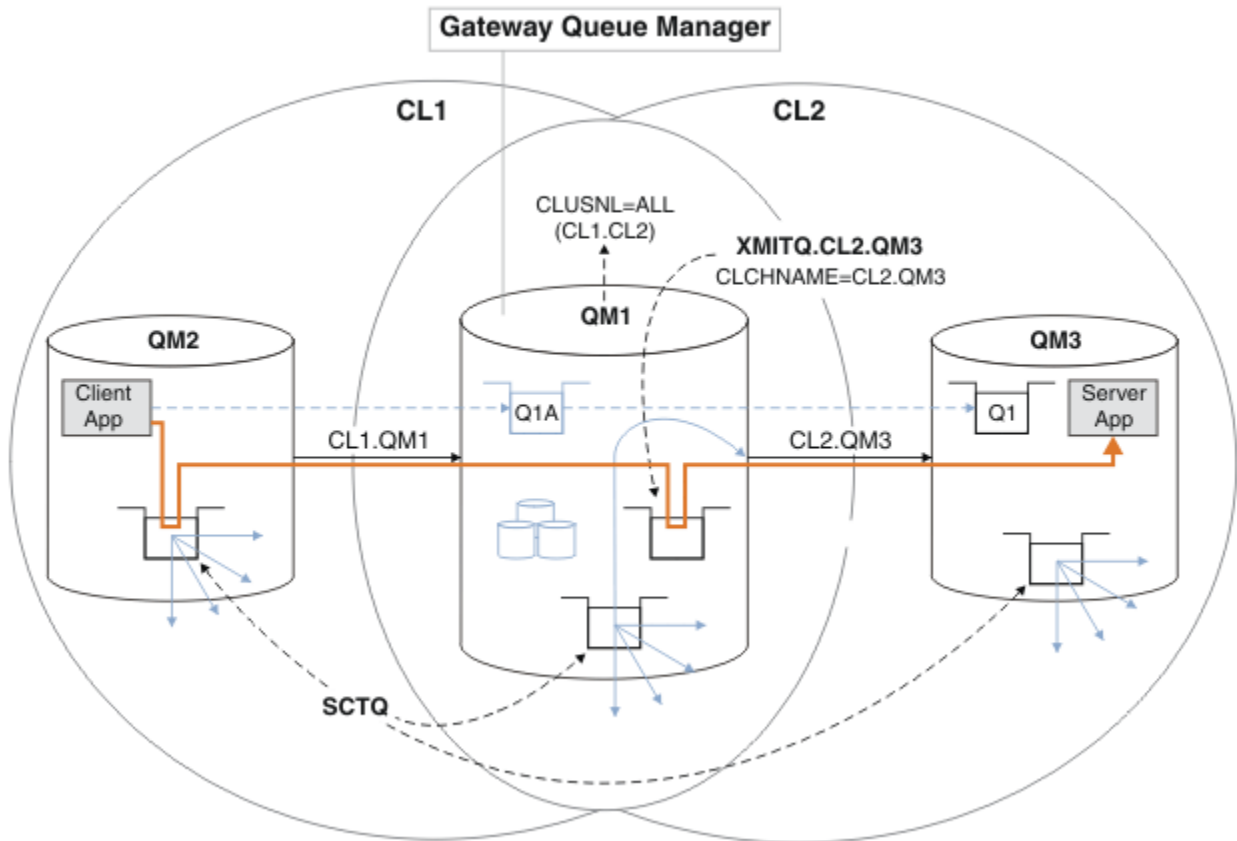


Figura 50. Applicazione client-server distribuita all'architettura hub e spoke utilizzando una coda di trasmissione cluster aggiuntiva.

Procedura

1. Creare un'ulteriore coda di trasmissione del cluster per il canale mittente del cluster CL2. QM3 sul gestore code del gateway, QM1.

```
*... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. Passare all'utilizzo della coda di trasmissione, XMITQ. CL2. QM3.
 - a) Arrestare il canale mittente del cluster CL2. QM3.

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

La risposta è che il comando è accettato:

```
AMQ8019: Stop IBM MQ channel accepted.
```

- b) Verificare che il canale CL2. QM3 sia arrestato

Se il canale non si arresta, è possibile eseguire nuovamente il comando **STOP CHANNEL** con l'opzione **FORCE**. Un esempio di impostazione dell'opzione **FORCE** è se il canale non si arresta e non è possibile riavviare l'altro gestore code per sincronizzare il canale.

```
*... On QM1
start
```

La risposta è un riepilogo dello stato del canale

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(STOPPED)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) Avviare il canale, CL2.QM3.

```
*... On QM1
START CHANNEL(CL2.QM3)
```

La risposta è che il comando è accettato:

```
AMQ8018: Start IBM MQ channel accepted.
```

d) Verificare che il canale sia stato avviato.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

La risposta è un riepilogo dello stato del canale:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(XMITQ.CL2.QM3)
```

e) Controllare che la coda di trasmissione sia stata commutata.

Monitorare il file di registrazione errori del gestore code del gateway per il messaggio " AMQ7341 La coda di trasmissione per il canale CL2.QM3 è XMITQ.CL2.QM3 ".

Operazioni successive

Verificare la coda di trasmissione separata inviando un messaggio da QM2 a Q1 on QM3 utilizzando la definizione dell'alias della coda Q1A

1. Eseguire il programma di esempio **amqsput** su QM2 per inserire un messaggio.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

2. Eseguire il programma di esempio **amqsget** per richiamare il messaggio da Q1 on QM3

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

Concetti correlati

Controllo accessi e code di trasmissione di più cluster

Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster

“Utilizzo delle code di trasmissione del cluster e dei canali mittente del cluster” a pagina 309

I messaggi tra i gestori code con cluster vengono memorizzati nelle code di trasmissione cluster e inoltrati dai canali mittenti del cluster. In qualsiasi momento, un canale mittente del cluster è associato a una coda di trasmissione. Se si modifica la configurazione del canale, questa potrebbe passare a una coda di trasmissione diversa al successivo avvio. L'elaborazione di questo switch è automatizzata e transazionale.

Attività correlate

Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una definizione remota della coda cluster e un canale mittente e una coda di trasmissione separati.

Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza un cluster aggiuntivo per isolare i messaggi in una particolare coda cluster.

Modifica del valore predefinito per separare le code di trasmissione del cluster per isolare il traffico dei messaggi

È possibile modificare il modo predefinito in cui un gestore code memorizza i messaggi per una coda cluster o un argomento su una coda di trasmissione. La modifica del valore predefinito fornisce un modo per isolare i messaggi cluster su un gestore code del gateway.

Clustering: pianificazione della configurazione delle code di trasmissione del cluster

“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza un cluster aggiuntivo per isolare i messaggi in una particolare coda cluster.

Prima di iniziare

I passi nell'attività ... vengono scritti per modificare la configurazione illustrata in [Figura 50 a pagina 360](#).

1. Il gestore code del gateway deve essere su IBM MQ.
2. Crea i cluster di sovrapposizione mostrati nell' [applicazione client - server distribuita all'architettura hub e spoke utilizzando i cluster IBM MQ in “Creazione di due cluster sovrapposti con un gestore code del gateway” a pagina 352](#) seguendo i passi in tale attività.

3. Eseguire i passi in [Figura 50 a pagina 360](#) in [“Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 359](#) per creare la soluzione senza il cluster aggiuntivo. Utilizzarlo come base per i passaggi in questa attività.

Informazioni su questa attività

La soluzione per isolare il traffico di messaggi a una singola applicazione in [“Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 359](#) funziona se la coda del cluster di destinazione è l'unica coda del cluster su un gestore code. Se non lo è, hai due scelte. Spostare la coda in un gestore code differente oppure creare un cluster che isoli la coda da altre code cluster sul gestore code.

Questa attività consente di aggiungere un cluster per isolare la coda di destinazione. Il cluster viene aggiunto solo per tale scopo. In pratica, affrontare l'attività di isolare sistematicamente alcune applicazioni quando si stanno progettando i cluster e gli schemi di denominazione dei cluster. L'aggiunta di un cluster ogni volta che una coda richiede l'isolamento potrebbe finire con molti cluster da gestire. In questa attività, si modifica la configurazione in [“Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 359](#) aggiungendo un cluster CL3 per isolare Q1 su QM3. Le applicazioni continuano ad essere eseguite per tutta la durata della modifica.

Le nuove definizioni e quelle modificate vengono evidenziate in [Figura 51 a pagina 364](#). Il riepilogo delle modifiche è il seguente: creare un cluster, il che significa che è necessario creare anche un nuovo repository cluster completo. Nell'esempio, QM3 è uno dei repository completi per CL3. Creare i canali mittente cluster e ricevente cluster per QM1 per aggiungere il gestore code gateway al nuovo cluster. Modificare la definizione di Q1 per passare a CL3. Modificare l'elenco nomi del cluster sul gestore code del gateway e aggiungere una coda di trasmissione del cluster per utilizzare il nuovo canale cluster. Infine, passare l'alias della coda Q1A al nuovo elenco nomi cluster.

IBM MQ non può trasferire automaticamente i messaggi dalla coda di trasmissione XMITQ . CL2 . QM3 aggiunta in [“Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway” a pagina 359](#) alla nuova coda di trasmissione XMITQ . CL3 . QM3. Può trasferire automaticamente i messaggi solo se entrambe le code di trasmissione sono servite dallo stesso canale mittente del cluster. Invece, l'attività descrive un modo per eseguire lo switch manualmente, che potrebbe essere appropriato per l'utente. Una volta completato il trasferimento, è possibile ripristinare l'uso della coda di trasmissione del cluster predefinita per altre code del cluster CL2 su QM3. In alternativa, è possibile continuare a utilizzare XMITQ . CL2 . QM3. Se si decide di ripristinare una coda di trasmissione del cluster predefinita, il gestore code del gateway gestisce automaticamente lo switch.

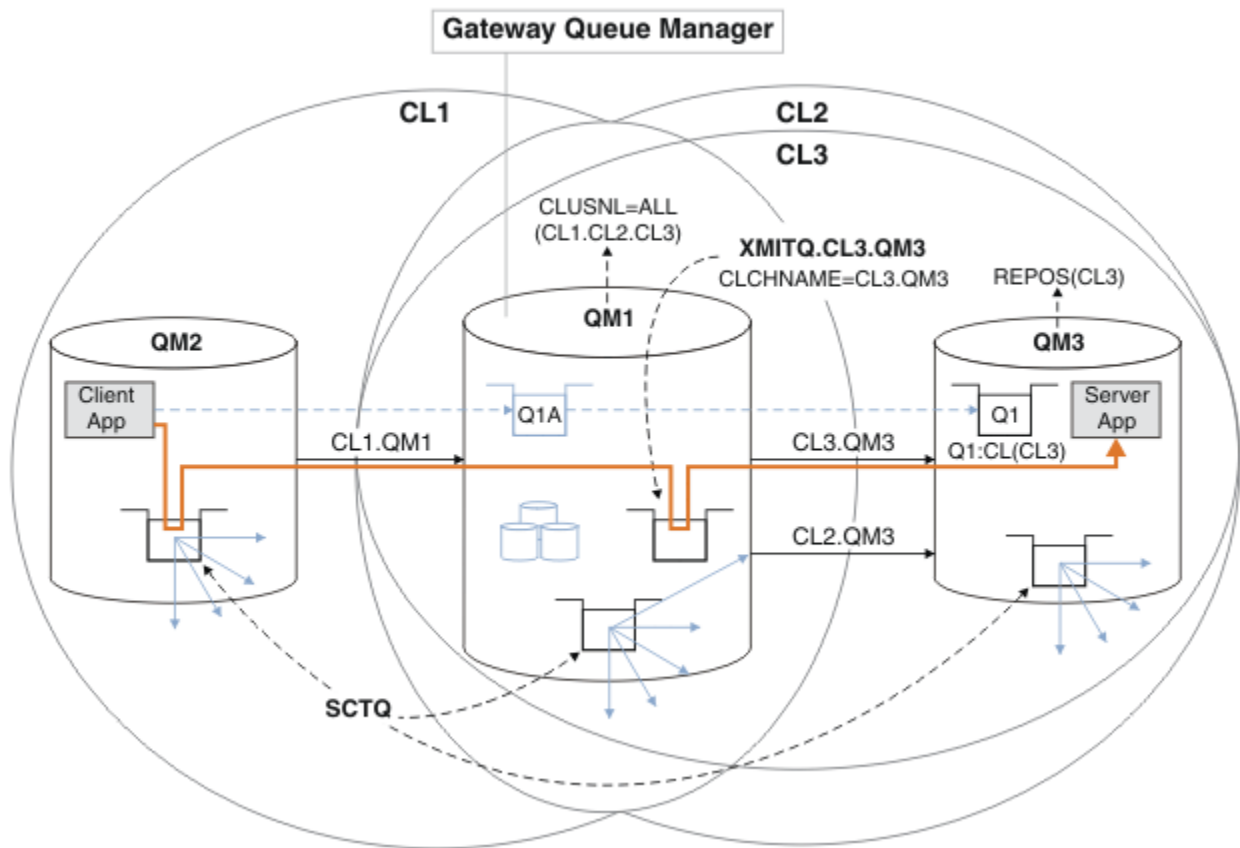


Figura 51. Utilizzo di un ulteriore cluster per separare il traffico di messaggi nel gestore code del gateway che va a una delle diverse code del cluster sullo stesso gestore code

Procedura

1. Modificare i gestori code QM3 e QM5 per renderli repository sia per CL2 che per CL3.

Per rendere un gestore code membro di più cluster, deve utilizzare un elenco di nomi cluster per identificare i cluster di cui è membro.

```
*... On QM3 and QM5
DEFINE NAMLIST(CL23) NAMES(CL2, CL3) REPLACE
ALTER QMGR REPOS(' ') REPOSNL(CL23)
```

2. Definire i canali tra i gestori code QM3 e QM5 per CL3.

```
*... On QM3
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

*... On QM5
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
```

3. Aggiungere il gestore code del gateway a CL3.

Aggiungere il gestore code del gateway aggiungendo QM1 a CL3 come repository parziale. Creare un repository parziale aggiungendo i canali mittente cluster e ricevente cluster a QM1.

Aggiungere inoltre CL3 all'elenco dei nomi di tutti i cluster connessi al gestore code del gateway.

```
*... On QM1
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMLIST(ALL) NAMES(CL1, CL2, CL3)
```

4. Aggiungere una coda di trasmissione cluster al gestore code del gateway, QM1, per i messaggi che vanno a CL3 su QM3.

Inizialmente, arrestare il canale mittente del cluster che trasferisce i messaggi dalla coda di trasmissione finché non si è pronti a commutare le code di trasmissione.

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

5. Eliminare i messaggi dalla coda di trasmissione del cluster esistente XMITQ.CL2.QM3.

Questa sottoprocedura è intesa a preservare l'ordine dei messaggi in Q1 in modo che corrisponda all'ordine in cui sono arrivati al gestore code del gateway. Con i cluster, l'ordinamento dei messaggi non è completamente garantito, ma è probabile. Se è richiesto l'ordine garantito dei messaggi, le applicazioni devono definirne l'ordine. Consultare [L'ordine in cui i messaggi vengono richiamati da una coda](#).

- a) Modificare la coda di destinazione Q1 su QM3 da CL2 a CL3.

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

- b) Monitorare XMITQ.CL3.QM3 fino a quando i messaggi non iniziano ad essere consegnati.

I messaggi vengono consegnati a XMITQ.CL3.QM3 quando il passaggio di Q1 a CL3 viene propagato al gestore code del gateway.

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
```

- c) Monitorare XMITQ.CL2.QM3 fino a quando non ha alcun messaggio in attesa di essere consegnato a Q1 su QM3.

Nota: XMITQ.CL2.QM3 potrebbe memorizzare messaggi per altre code su QM3 che sono membri di CL2, nel qual caso la profondità potrebbe non andare a zero.

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
```

- d) Abilitare il richiamo dalla nuova coda di trasmissione del cluster, XMITQ.CL3.QM3

```
*... On QM1
ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
```

6. Rimuovere la vecchia coda di trasmissione del cluster, XMITQ.CL2.QM3, se non è più richiesta.

I messaggi per le code del cluster in CL2 su QM3 ritornano all'utilizzo della coda di trasmissione del cluster predefinita sul gestore code del gateway, QM1. La coda di trasmissione cluster predefinita è SYSTEM.CLUSTER.TRANSMIT.QUEUE o SYSTEM.CLUSTER.TRANSMIT.CL2.QM3. La scelta dipende dal fatto che il valore dell'attributo del gestore code **DEFCLXQ** su QM1 sia SCTQ o CHANNEL. Il gestore code trasferisce i messaggi da XMITQ.CL2.QM3 automaticamente al successivo avvio del canale mittente del cluster CL2.QM3.

- a) Modificare la coda di trasmissione, XMITQ.CL2.QM3, da coda di trasmissione cluster a coda di trasmissione normale.

Ciò interrompe l'associazione della coda di trasmissione con qualsiasi canale mittente del cluster. In risposta, IBM MQ trasferisce automaticamente i messaggi da XMITQ.CL2.QM3 alla coda di trasmissione del cluster predefinita al successivo avvio del canale mittente del cluster. Fino a quel momento, i messaggi per CL2 su QM3 continuano a essere posizionati su XMITQ.CL2.QM3.

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME('')
```

b) Arrestare il canale mittente del cluster CL2.QM3.

L'arresto e il riavvio del canale mittente del cluster avvia il trasferimento dei messaggi da XMITQ.CL2.QM3 nella coda di trasmissione del cluster predefinita. In genere si arresta e si avvia il canale manualmente per avviare il trasferimento. Il trasferimento viene avviato automaticamente se il canale viene riavviato dopo la chiusura alla scadenza del suo intervallo di disconnessione.

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

La risposta è che il comando è accettato:

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) Verificare che il canale CL2.QM3 sia arrestato

Se il canale non si arresta, è possibile eseguire nuovamente il comando **STOP CHANNEL** con l'opzione **FORCE**. Un esempio di impostazione dell'opzione **FORCE** è se il canale non si arresta e non è possibile riavviare l'altro gestore code per sincronizzare il canale.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

La risposta è un riepilogo dello stato del canale

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                   STATUS(STOPPED)
SUBSTATE(MQGET)                XMITQ(XMITQ.CL2.QM3)
```

d) Avviare il canale, CL2.QM3.

```
*... On QM1
START CHANNEL(CL2.QM3)
```

La risposta è che il comando è accettato:

```
AMQ8018: Start IBM MQ channel accepted.
```

e) Verificare che il canale sia stato avviato.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

La risposta è un riepilogo dello stato del canale:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                   STATUS(RUNNING)
SUBSTATE(MQGET)                XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE|CL2.QM3)
```

f) Monitorare il file di registrazione errori del gestore code del gateway per il messaggio " AMQ7341 La coda di trasmissione per il canale CL2.QM3 è SYSTEM.CLUSTER.TRANSMIT.QUEUE|CL2.QM3 ".

g) Eliminare la coda di trasmissione cluster, XMITQ.CL2.QM3.

```
*... On QM1  
DELETE QLOCAL(XMITQ.CL2.QM3)
```

Operazioni successive

Verificare la coda con cluster separata inviando un messaggio da QM2 a Q1 su QM3 utilizzando la definizione alias della coda Q1A

1. Eseguire il programma di esempio **amqspu**t su QM2 per inserire un messaggio.

```
C:\IBM\MQ>amqspu Q1A QM2  
Sample AMQSPUT0 start  
target queue is Q1A  
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. Eseguire il programma di esempio **amqsget** per richiamare il messaggio da Q1 on QM3

```
C:\IBM\MQ>amqsget Q1 QM3  
Sample AMQSGET0 start  
message <Sample request message from QM2 to Q1 using Q1A>  
no more messages  
Sample AMQSGET0 end
```

Concetti correlati

Controllo accessi e code di trasmissione di più cluster

Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster

“Utilizzo delle code di trasmissione del cluster e dei canali mittente del cluster” a pagina 309

I messaggi tra i gestori code con cluster vengono memorizzati nelle code di trasmissione cluster e inoltrati dai canali mittenti del cluster. In qualsiasi momento, un canale mittente del cluster è associato a una coda di trasmissione. Se si modifica la configurazione del canale, questa potrebbe passare a una coda di trasmissione diversa al successivo avvio. L'elaborazione di questo switch è automatizzata e transazionale.

Attività correlate

Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway
Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una definizione remota della coda cluster e un canale mittente e una coda di trasmissione separati.

Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una coda di trasmissione cluster aggiuntiva per separare il traffico di messaggi a un singolo gestore code in un cluster.

Modifica del valore predefinito per separare le code di trasmissione del cluster per isolare il traffico dei messaggi

È possibile modificare il modo predefinito in cui un gestore code memorizza i messaggi per una coda cluster o un argomento su una coda di trasmissione. La modifica del valore predefinito fornisce un modo per isolare i messaggi cluster su un gestore code del gateway.

Clustering: pianificazione della configurazione delle code di trasmissione del cluster

“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Modifica del valore predefinito per separare le code di trasmissione del cluster per isolare il traffico dei messaggi

È possibile modificare il modo predefinito in cui un gestore code memorizza i messaggi per una coda cluster o un argomento su una coda di trasmissione. La modifica del valore predefinito fornisce un modo per isolare i messaggi cluster su un gestore code del gateway.

Prima di iniziare

1. Il gestore code del gateway deve essere su IBM MQ.
2. Crea i cluster di sovrapposizione mostrati nell' [applicazione client - server distribuita all'architettura hub e spoke utilizzando i cluster IBM MQ in “Creazione di due cluster sovrapposti con un gestore code del gateway” a pagina 352](#) seguendo i passi in tale attività.

Informazioni su questa attività

Per implementare l'architettura con più code cluster, il tuo gestore code gateway deve essere su IBM MQ. Per utilizzare più code di trasmissione del cluster, è necessario modificare il tipo di coda di trasmissione del cluster predefinito sul gestore code del gateway. Modificare il valore dell'attributo del gestore code **DEFCLXQ** su QM1 da SCTQ a CHANNEL ; consultare [Figura 52 a pagina 369](#). Il diagramma mostra un flusso di messaggi. Per i flussi verso altri gestori code o verso altri cluster, il gestore code crea ulteriori code di trasmissione del cluster dinamico permanenti. Ogni canale mittente del cluster trasferisce i messaggi da una coda di trasmissione cluster differente.

La modifica non ha effetto immediato, a meno che non si stia connettendo il gestore code del gateway ai cluster per la prima volta. L'attività include i passi per il caso tipico di gestione di una modifica a una configurazione esistente. Per impostare un gestore code in modo che utilizzi code di trasmissione cluster separate quando si unisce per la prima volta a un cluster; consultare [“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331](#).

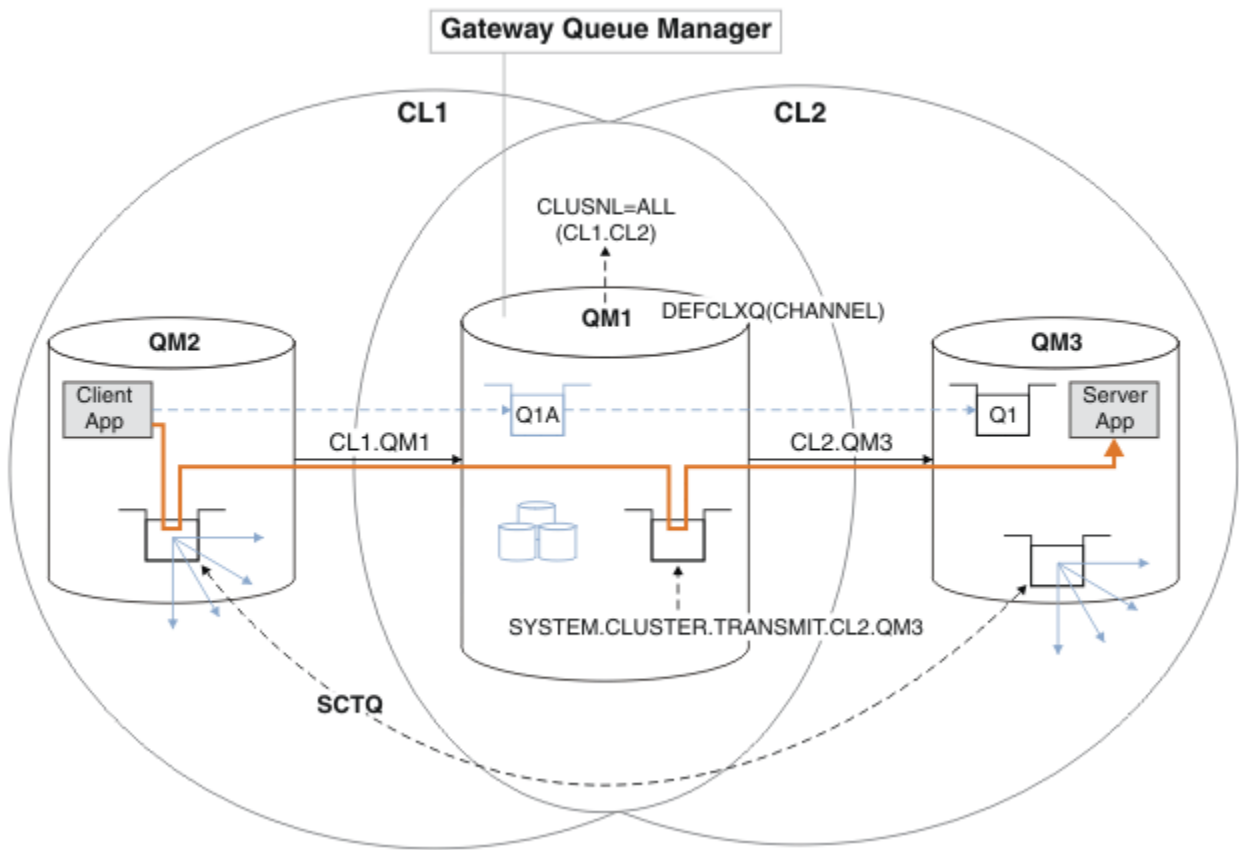


Figura 52. Applicazione client-server distribuita all'architettura hub e spoke con code di trasmissione cluster separate sul gestore code del gateway.

Procedura

1. Modificare il gestore code del gateway in modo da utilizzare code di trasmissione cluster separate.

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. Passare alle code di trasmissione cluster separate.

Qualsiasi canale mittente del cluster che non è in esecuzione passa all'utilizzo di code di trasmissione del cluster separate al successivo avvio.

Per commutare i canali in esecuzione, riavviare il gestore code oppure attenersi alla seguente procedura:

- a) Elencare i canali mittente del cluster in esecuzione con `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

La risposta è un elenco di report di stato del canale:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1412))       CURRENT
RQMNAME(QM2)                    STATUS(RUNNING)
SUBSTATE(MQGET)                 XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
```

```

CHANNEL (CL2.QM3)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)             STATUS (RUNNING)
SUBSTATE (MQGET)         XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM5)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415)) CURRENT
RQMNAME (QM5)             STATUS (RUNNING)
SUBSTATE (MQGET)         XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
RQMNAME (QM4)             STATUS (RUNNING)
SUBSTATE (MQGET)         XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

b) Arresta i canali in esecuzione

Per ogni canale nell'elenco, eseguire il comando:

```

*... On QM1
STOP CHANNEL (ChannelName)

```

Dove *ChannelName* è ognuno di CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5.

La risposta è che il comando è accettato:

AMQ8019: Stop IBM MQ channel accepted.

c) Monitorare quali canali sono arrestati

```

*... On QM1
DISPLAY CHSTATUS(*) WHERE (XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')

```

La risposta è un elenco di canali ancora in esecuzione e di canali arrestati:

```

AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM2)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1412)) CURRENT
RQMNAME (QM2)             STATUS (STOPPED)
SUBSTATE ( )              XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)             STATUS (STOPPED)
SUBSTATE ( )              XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM5)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415)) CURRENT
RQMNAME (QM5)             STATUS (STOPPED)
SUBSTATE ( )              XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
RQMNAME (QM4)             STATUS (STOPPED)
SUBSTATE ( )              XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

d) Avviare ogni canale arrestato.

Eseguire questa operazione per tutti i canali in esecuzione. Se un canale non si arresta, è possibile eseguire nuovamente il comando **STOP CHANNEL** con l'opzione **FORCE**. Un esempio

di impostazione dell'opzione FORCE è se il canale non si arresta e non è possibile riavviare l'altro gestore code per sincronizzare il canale.

```
*... On QM1
START CHANNEL(CL2.QM5)
```

La risposta è che il comando è accettato:

AMQ8018: Start IBM MQ channel accepted.

e) Monitorare le code di trasmissione che vengono commutate.

Monitorare il file di registrazione errori del gestore code del gateway per il messaggio " AMQ7341
La coda di trasmissione per il canale CL2.QM3 è SYSTEM.CLUSTER.TRANSMIT.
QUEUE/CL2.QM3 ".

f) Verificare che SYSTEM.CLUSTER.TRANSMIT.QUEUE non sia più utilizzato

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

La risposta è un elenco di report di stato del canale e la profondità di
SYSTEM.CLUSTER.TRANSMIT.QUEUE:

```
AMQ8420: Channel Status not found.
AMQ8409: Display Queue details.
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)      TYPE(QLOCAL)
CURDEPTH(0)
```

g) Monitorare quali canali sono avviati

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

La risposta è un elenco dei canali, in questo caso già in esecuzione con le nuove code di
trasmissione del cluster predefinite:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)                                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1412))                       CURRENT
RQMNAME(QM2)                                     STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))                       CURRENT
RQMNAME(QM3)                                     STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)                                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1415))                       CURRENT
RQMNAME(QM5)                                     STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)                                CHLTYPE(CLUSSDR)
```

```
CONNNAME(127.0.0.1(1414))          CURRENT
RQMNAME(QM4)                       STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

Operazioni successive

1. Verificare la coda di trasmissione del cluster definita automaticamente inviando un messaggio da QM2 a Q1 on QM3, risolvendo il nome della coda con definizione dell'alias della coda Q1A
 - a. Eseguire il programma di esempio **amqsput** su QM2 per inserire un messaggio.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```

- b. Eseguire il programma di esempio **amqsget** per richiamare il messaggio da Q1 on QM3

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. Considerare se riconfigurare la sicurezza, configurando la sicurezza per le code del cluster sui gestori code in cui hanno origine i messaggi per le code del cluster.

Concetti correlati

Controllo accessi e code di trasmissione di più cluster

Clustering: isolamento dell'applicazione utilizzando più code di trasmissione cluster

Attività correlate

Aggiunta di una definizione di coda remota per isolare i messaggi inviati da un gestore code del gateway
Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una definizione remota della coda cluster e un canale mittente e una coda di trasmissione separati.

Aggiunta di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza una coda di trasmissione cluster aggiuntiva per separare il traffico di messaggi a un singolo gestore code in un cluster.

Aggiunta di un cluster e di una coda di trasmissione cluster per isolare il traffico di messaggi cluster inviati da un gestore code gateway

Modificare la configurazione dei cluster sovrapposti che utilizzano un gestore code gateway. Dopo che i messaggi di modifica sono stati trasferiti a un'applicazione dal gestore code del gateway senza utilizzare la stessa coda di trasmissione o gli stessi canali degli altri messaggi cluster. La soluzione utilizza un cluster aggiuntivo per isolare i messaggi in una particolare coda cluster.

Clustering: pianificazione della configurazione delle code di trasmissione del cluster

“Aggiunta di un gestore code a un cluster: code di trasmissione separate” a pagina 331

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code cluster e gli argomenti vengono trasferiti utilizzando più code di trasmissione cluster.

Rimozione di una coda cluster da un gestore code

Disabilitare la coda INVENTQ a Toronto. Inviare tutti i messaggi di inventario a New York ed eliminare la coda INVENTQ a Toronto quando è vuota.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in [“Aggiunta di un gestore code su cui è presente una coda”](#) a pagina 336. Contiene quattro gestori code. LONDON e NEWYORK contengono entrambi repository completi. PARIS e TORONTO contengono repository parziali. L'applicazione di inventario viene eseguita sui sistemi di New York e Toronto ed è guidata dall'arrivo dei messaggi sulla coda INVENTQ .
- A causa della riduzione del carico di lavoro, non si desidera più eseguire l'applicazione di inventario a Toronto. Si desidera disabilitare la coda INVENTQ ospitata dal gestore code TORONTO e disporre di messaggi feed TORONTO nella coda INVENTQ in NEWYORK.
- La connettività di rete esiste tra tutti e quattro i sistemi.
- Il protocollo di rete è TCP.

Informazioni su questa attività

Effettuare le operazioni riportate di seguito per rimuovere una coda cluster.

Procedura

1. Indica che la coda non è più disponibile.

Per rimuovere una coda da un cluster, rimuovere il nome cluster dalla definizione della coda locale. Modificare il INVENTQ su TORONTO in modo che non sia accessibile dal resto del cluster:

```
ALTER QLOCAL(INVENTQ) CLUSTER('')
```

2. Verificare che la coda non sia più disponibile.

Su un gestore code del repository completo, LONDON o NEWYORK, verificare che la coda non sia più ospitata dal gestore code TORONTO emettendo il seguente comando:

```
DIS QCLUSTER (INVENTQ)
```

TORONTO non è elencato nei risultati, se il comando ALTER è stato completato correttamente.

3. Disabilitare la coda.

Disabilitare la coda INVENTQ in TORONTO in modo che non sia possibile scrivere ulteriori messaggi:

```
ALTER QLOCAL(INVENTQ) PUT(DISABLED)
```

Ora i messaggi in transito verso questa coda utilizzando MQ00_BIND_ON_OPEN vanno alla coda di messaggi non instradabili. È necessario impedire a tutte le applicazioni di inserire esplicitamente i messaggi nella coda su questo gestore code.

4. Monitorare la coda finché non è vuota.

Monitorare la coda utilizzando il comando DISPLAY QUEUE , specificando gli attributi IPPROCS, OPPROCS e CURDEPTH oppure utilizzare il comando **WRKMQMSTS** su IBM i. Quando il numero di processi di input e di output e la profondità corrente delle code sono tutti zero, la coda è vuota.

5. Monitorare il canale per assicurarsi che non vi siano messaggi in dubbio.

Per essere certi che non vi siano messaggi in dubbio sul canale INVENTORY . TORONTO, monitorare il canale mittente del cluster denominato INVENTORY . TORONTO su ciascuno degli altri gestori code. Immettere il comando DISPLAY CHSTATUS specificando il parametro INDOUBT da ogni gestore code:

```
DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
```

Se sono presenti messaggi in dubbio, è necessario risolverli prima di procedere. Ad esempio, è possibile provare ad emettere il comando di canale RESOLVE o ad arrestare e riavviare il canale.

6. Eliminare la coda locale.

Quando si è soddisfatti che non ci sono più messaggi da consegnare all'applicazione di inventario in TORONTO, è possibile eliminare la coda:

```
DELETE QLOCAL(INVENTQ)
```

7. Ora è possibile rimuovere l'applicazione di inventario dal sistema a Toronto

La rimozione dell'applicazione evita la duplicazione e consente di risparmiare spazio sul sistema.

Risultati

Il cluster impostato da questa attività è simile a quello impostato dall'attività precedente. La differenza è che la coda INVENTQ non è più disponibile sul gestore code TORONTO.

Quando la coda è stata tolta dal servizio nel passo 1, il gestore code TORONTO ha inviato un messaggio ai due gestori code del repository completo. Ha notificato loro la modifica dello stato. I gestori code del repository completo trasmettono queste informazioni ad altri gestori code del cluster che hanno richiesto aggiornamenti alle informazioni relative a INVENTQ.

Quando un gestore code inserisce un messaggio nella coda INVENTQ, il repository parziale aggiornato indica che la coda INVENTQ è disponibile solo sul gestore code NEWYORK. Il messaggio viene inviato al gestore code NEWYORK.

Operazioni successive

In questa attività, c'era solo una coda da rimuovere e solo un cluster da cui rimuoverla.

Si supponga che vi siano molte code che fanno riferimento a un elenco nomi contenente molti nomi cluster. Ad esempio, il gestore code TORONTO potrebbe contenere non solo INVENTQ, ma anche PAYROLLQ, SALESQ e PURCHASESQ. TORONTO rende queste code disponibili in tutti i cluster appropriati, INVENTORY, PAYROLL, SALES e PURCHASES. Definire un elenco nomi dei nomi cluster sul gestore code TORONTO:

```
DEFINE NAMELIST(TOROLIST)  
DESCR('List of clusters TORONTO is in')  
NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

Aggiungere l'elenco nomi a ciascuna definizione di coda:

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)  
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)  
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)  
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

Ora si supponga di voler rimuovere tutte le code dal cluster SALES, perché l'operazione SALES deve essere presa in consegna dall'operazione PURCHASES. Tutto ciò che devi fare è modificare l'elenco nomi TOROLIST per rimuovere il nome del cluster SALES da esso.

Se si desidera rimuovere una singola coda da uno dei cluster nell'elenco nomi, creare un elenco nomi contenente il rimanente elenco di nomi cluster. Quindi, modificare la definizione della coda per utilizzare il nuovo elenco nomi. Per rimuovere PAYROLLQ dal cluster INVENTORY :

1. Creare un elenco nomi:

```
DEFINE NAMELIST(TOROSHORTLIST)
DESCR('List of clusters TORONTO is in other than INVENTORY')
NAMES(PAYROLL, SALES, PURCHASES)
```

2. Modificare la definizione della coda PAYROLLQ :

```
ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
```

Rimozione di un gestore code da un cluster: procedura ottimale

Rimuovere un gestore code da un cluster, in scenari in cui il gestore code può comunicare normalmente con almeno un repository completo nel cluster.

Prima di iniziare

Questo metodo è la procedura ottimale per gli scenari in cui è disponibile almeno un repository completo e può essere contattato dal gestore code che viene rimosso. Questo metodo implica il minimo intervento manuale e consente al gestore code di negoziare un ritiro controllato dal cluster. Se il gestore code che viene rimosso non può contattare un repository completo, consultare [“Rimozione di un gestore code da un cluster: metodo alternativo” a pagina 377](#).

Informazioni su questa attività

Questa attività di esempio rimuove il gestore code LONDON dal cluster INVENTORY . Il cluster INVENTORY è impostato come descritto in [“Aggiunta di un gestore code a un cluster” a pagina 329](#) e modificato come descritto in [“Rimozione di una coda cluster da un gestore code” a pagina 373](#).

Il processo di rimozione di un gestore code da un cluster è più complicato del processo di aggiunta di un gestore code.

Quando un gestore code si unisce a un cluster, i membri esistenti del cluster non conoscono il nuovo gestore code e quindi non hanno interazioni con esso. È necessario creare nuovi canali mittente e ricevente sul gestore code di unione in modo che possa connettersi a un repository completo.

Quando un gestore code viene rimosso da un cluster, è probabile che le applicazioni connesse al gestore code utilizzino oggetti come code ospitate altrove nel cluster. Inoltre, le applicazioni connesse ad altri gestori code nel cluster potrebbero utilizzare oggetti ospitati sul gestore code di destinazione. Come risultato di queste applicazioni, il gestore code corrente potrebbe creare ulteriori canali mittente per stabilire la comunicazione con i membri del cluster diversi dal repository completo utilizzato per unirsi al cluster. Ogni gestore code nel cluster dispone di una copia memorizzata nella cache dei dati che descrive altri membri cluster. Ciò potrebbe includere quello che si sta rimuovendo.

Procedura

1. Prima di rimuovere il gestore code dal cluster, verificare che il gestore code non ospiti più le risorse richieste dal cluster:
 - Se il gestore code ospita un repository completo, completare i passi da 1 a 6 da [“Spostamento di un repository completo in un altro gestore code” a pagina 340](#). Se la funzionalità del repository completo del gestore code da rimuovere non deve essere spostata in un altro gestore code, è necessario solo completare i passi 5 e 6.
 - Se il gestore code ospita le code cluster, completare i passi da 1 a 7 da [“Rimozione di una coda cluster da un gestore code” a pagina 373](#).

- Se il gestore code ospita argomenti cluster, eliminare gli argomenti (ad esempio utilizzando il comando `DELETE TOPIC`) o spostarli su altri host come descritto in [“Spostamento di una definizione di argomento del raggruppamento in un gestore code differente”](#) a pagina 456.

Nota: Se si rimuove un gestore code da un cluster e il gestore code ospita ancora un argomento cluster, il gestore code potrebbe continuare a tentare di consegnare le pubblicazioni ai gestori code rimasti nel cluster fino a quando l'argomento non viene eliminato.

2. Modificare i canali riceventi del cluster definiti manualmente per rimuoverli dal cluster, sul gestore code LONDON:

```
ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

3. Modificare i canali mittenti del cluster definiti manualmente per rimuoverli dal cluster sul gestore code LONDON:

```
ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
```

Gli altri gestori code del cluster apprendono che questo gestore code e le sue risorse cluster non fanno più parte del cluster.

4. Monitorare la coda di trasmissione del cluster, sul gestore code LONDON, fino a quando non sono presenti messaggi in attesa di essere trasmessi a un repository completo nel cluster.

```
DISPLAY CHSTATUS(INVENTORY.PARIS) XQMSGSA
```

Se i messaggi rimangono nella coda di trasmissione, determinare il motivo per cui non vengono inviati ai repository completi PARIS e NEWYORK prima di continuare.

Risultati

Il gestore code LONDON non è più parte del cluster. Tuttavia, può ancora funzionare come gestore code indipendente.

Operazioni successive

Il risultato di queste modifiche può essere confermato emettendo il seguente comando sui restanti membri del cluster:

```
DISPLAY CLUSQMGR(LONDON)
```

Il gestore code continua ad essere visualizzato fino a quando i canali mittenti del cluster definiti automaticamente non vengono arrestati. È possibile attendere che ciò si verifichi oppure continuare a monitorare le istanze attive immettendo il seguente comando:

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

Quando si è certi che nessun altro messaggio viene recapitato a questo gestore code, è possibile arrestare i canali mittenti del cluster in LONDON immettendo il seguente comando sui restanti membri del cluster:

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```


Dopo che le modifiche sono state propagate in tutto il cluster e che non sono stati consegnati ulteriori messaggi a questo gestore code, arrestare ed eliminare il canale CLUSRCVR su LONDON:

```
STOP CHANNEL (INVENTORY.LONDON)
DELETE CHANNEL (INVENTORY.LONDON)
```

Se per questo canale era in uso una coda di trasmissione definita manualmente e il pattern CLCHNAME non corrisponde ad altri canali esistenti o pianificati, è possibile eliminare la coda di trasmissione. Ad esempio:

```
DELETE QLOCAL (PARIS.CUSTOM.XMITQ)
```

Nota: Se le code di trasmissione definite automaticamente o il SYSTEM.CLUSTER.TRANSMIT.QUEUE è in uso, questo passo non è richiesto.

Il gestore code rimosso può essere aggiunto nuovamente al cluster in un secondo momento, come descritto in [“Aggiunta di un gestore code a un cluster”](#) a pagina 329. Il gestore code rimosso continua a memorizzare nella cache i membri rimanenti del cluster per un periodo massimo di 90 giorni. Se si preferisce non attendere la scadenza di questa cache, è possibile rimuoverla forzatamente come descritto in [“Ripristino di un gestore code allo stato pre - cluster”](#) a pagina 379.

Attività correlate

[Rimozione di un gestore code da un cluster \(utilizzando IBM MQ Explorer\)](#)

Riferimenti correlati

[ALTER CHANNEL \(modifica impostazioni canale\)](#)

[DISPLAY CHANNEL \(visualizzazione definizione canale\)](#)

[DISPLAY CHSTATUS \(visualizzazione stato canale\)](#)

[DISPLAY CLUSQMGR \(visualizza informazioni sul canale per gestori code cluster\)](#)

[STOP CHANNEL \(arresta canale\)](#)

Rimozione di un gestore code da un cluster: metodo alternativo

Rimuovere un gestore code da un cluster, in scenari in cui, a causa di un significativo problema di sistema o di configurazione, il gestore code non può comunicare con alcun repository completo nel cluster.

Prima di iniziare

Questo metodo alternativo di rimozione di un gestore code da un cluster arresta ed elimina manualmente tutti i canali cluster che collegano al cluster il gestore code rimosso e rimuove forzatamente il gestore code dal cluster. Questo metodo viene utilizzato in scenari in cui il gestore code che viene rimosso non può comunicare con nessuno dei repository completi. Ciò potrebbe verificarsi (ad esempio) perché il gestore code ha smesso di funzionare o perché si è verificato un errore di comunicazioni prolungato tra il gestore code e il cluster. Altrimenti, utilizzare il metodo più comune: [“Rimozione di un gestore code da un cluster: procedura ottimale”](#) a pagina 375.

Informazioni su questa attività

Questa attività di esempio rimuove il gestore code LONDON dal cluster INVENTORY. Il cluster INVENTORY è impostato come descritto in [“Aggiunta di un gestore code a un cluster”](#) a pagina 329 e modificato come descritto in [“Rimozione di una coda cluster da un gestore code”](#) a pagina 373.

Il processo di rimozione di un gestore code da un cluster è più complicato del processo di aggiunta di un gestore code.

Quando un gestore code si unisce a un cluster, i membri esistenti del cluster non conoscono il nuovo gestore code e quindi non hanno interazioni con esso. È necessario creare nuovi canali mittente e ricevente sul gestore code di unione in modo che possa connettersi a un repository completo.

Quando un gestore code viene rimosso da un cluster, è probabile che le applicazioni connesse al gestore code utilizzino oggetti come code ospitate altrove nel cluster. Inoltre, le applicazioni connesse ad altri gestori code nel cluster potrebbero utilizzare oggetti ospitati sul gestore code di destinazione. Come risultato di queste applicazioni, il gestore code corrente potrebbe creare ulteriori canali mittente per stabilire la comunicazione con i membri del cluster diversi dal repository completo utilizzato per unirsi al cluster. Ogni gestore code nel cluster dispone di una copia memorizzata nella cache dei dati che descrive altri membri cluster. Ciò potrebbe includere quello che si sta rimuovendo.

Questa procedura potrebbe essere appropriata in caso di emergenza, quando non è possibile attendere che il gestore code lasci il cluster correttamente.

Procedura

1. Prima di rimuovere il gestore code dal cluster, verificare che il gestore code non ospiti più le risorse richieste dal cluster:
 - Se il gestore code ospita un repository completo, completare i passi da 1 a 6 da [“Spostamento di un repository completo in un altro gestore code”](#) a pagina 340. Se la funzionalità del repository completo del gestore code da rimuovere non deve essere spostata in un altro gestore code, è necessario solo completare i passi 5 e 6.
 - Se il gestore code ospita le code cluster, completare i passi da 1 a 7 da [“Rimozione di una coda cluster da un gestore code”](#) a pagina 373.
 - Se il gestore code ospita argomenti cluster, eliminare gli argomenti (ad esempio utilizzando il comando DELETE TOPIC) o spostarli su altri host come descritto in [“Spostamento di una definizione di argomento del raggruppamento in un gestore code differente”](#) a pagina 456.

Nota: Se si rimuove un gestore code da un cluster e il gestore code ospita ancora un argomento cluster, il gestore code potrebbe continuare a tentare di consegnare le pubblicazioni ai gestori code rimasti nel cluster fino a quando l'argomento non viene eliminato.

2. Arresta tutti i canali utilizzati per comunicare con altri gestori code nel cluster. Utilizzare MODE (FORCE) per arrestare il canale di CLUSRCVR sul gestore code LONDON. Altrimenti, potrebbe essere necessario attendere che il gestore code mittente arresti il canale:

```
STOP CHANNEL (INVENTORY.LONDON) MODE (FORCE)
STOP CHANNEL (INVENTORY.TORONTO)
STOP CHANNEL (INVENTORY.PARIS)
STOP CHANNEL (INVENTORY.NEWYORK)
```

3. Monitorare gli stati del canale, nel gestore code LONDON, fino a quando i canali non vengono arrestati:

```
DISPLAY CHSTATUS (INVENTORY.LONDON)
DISPLAY CHSTATUS (INVENTORY.TORONTO)
DISPLAY CHSTATUS (INVENTORY.PARIS)
DISPLAY CHSTATUS (INVENTORY.NEWYORK)
```

Nessun altro messaggio dell'applicazione viene inviato a o dagli altri gestori code nel cluster dopo l'arresto dei canali.

4. Eliminare i canali cluster definiti manualmente, sul gestore code LONDON:

```
DELETE CHANNEL (INVENTORY.NEWYORK)
DELETE CHANNEL (INVENTORY.TORONTO)
```

5. I restanti gestori code nel cluster conservano ancora la conoscenza del gestore code rimosso e potrebbero continuare a inviargli messaggi. Per eliminare la conoscenza dai rimanenti gestori code, reimpostare il gestore code rimosso dal cluster su uno dei repository completi:

```
RESET CLUSTER (INVENTORY) ACTION (FORCEREMOVE) QMNAME (LONDON) QUEUES (YES)
```

Se nel cluster potrebbe essere presente un altro gestore code con lo stesso nome del gestore code rimosso, specificare il **QMID** del gestore code rimosso.

Risultati

Il gestore code LONDON non è più parte del cluster. Tuttavia, può ancora funzionare come gestore code indipendente.

Operazioni successive

Il risultato di queste modifiche può essere confermato emettendo il seguente comando sui restanti membri del cluster:

```
DISPLAY CLUSQMGR(LONDON)
```

Il gestore code continua ad essere visualizzato fino a quando i canali mittenti del cluster definiti automaticamente non vengono arrestati. È possibile attendere che ciò si verifichi oppure continuare a monitorare le istanze attive immettendo il seguente comando:

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

Dopo che le modifiche sono state propagate in tutto il cluster e che non sono stati recapitati ulteriori messaggi a questo gestore code, eliminare il canale CLUSRCVR su LONDON:

```
DELETE CHANNEL(INVENTORY.LONDON)
```

Il gestore code rimosso può essere aggiunto nuovamente al cluster in un secondo momento, come descritto in [“Aggiunta di un gestore code a un cluster”](#) a pagina 329. Il gestore code rimosso continua a memorizzare nella cache i membri rimanenti del cluster per un periodo massimo di 90 giorni. Se si preferisce non attendere la scadenza di questa cache, è possibile rimuoverla forzatamente come descritto in [“Ripristino di un gestore code allo stato pre - cluster”](#) a pagina 379.

Riferimenti correlati

[DELETE CHANNEL \(elimina un canale\)](#)

[DISPLAY CHANNEL \(visualizzazione definizione canale\)](#)

[DISPLAY CHSTATUS \(visualizzazione stato canale\)](#)

[DISPLAY CLUSQMGR \(visualizza informazioni sul canale per gestori code cluster\)](#)

[STOP CHANNEL \(arresta canale\)](#)

[RESET CLUSTER \(ripristino di un cluster\)](#)

Ripristino di un gestore code allo stato pre - cluster

Quando un gestore code viene rimosso da un cluster, conserva la conoscenza dei restanti membri del cluster. Questa conoscenza alla fine scade e viene eliminata automaticamente. Tuttavia, se si preferisce eliminarlo immediatamente, è possibile utilizzare i passi riportati in questo argomento.

Prima di iniziare

Si presume che il gestore code sia stato rimosso dal cluster e che non stia più eseguendo alcuna attività nel cluster. Ad esempio, le code non ricevono più messaggi dal cluster e nessuna applicazione è in attesa che i messaggi arrivino in queste code.

Informazioni su questa attività

Quando un gestore code viene rimosso da un cluster, conserva la conoscenza dei restanti membri del cluster per un massimo di 90 giorni. Ciò può avere dei vantaggi di sistema, in particolare se il gestore code si unisce rapidamente al cluster. Quando questa conoscenza alla fine scade, viene eliminata automaticamente. Tuttavia, ci sono dei motivi per cui è preferibile eliminare queste informazioni manualmente. Ad esempio:

- È possibile confermare di aver arrestato tutte le applicazioni su questo gestore code che in precedenza utilizzavano le risorse cluster. Fino alla scadenza della conoscenza dei restanti membri del cluster, qualsiasi applicazione continua a scrivere in una coda di trasmissione. Una volta eliminata la conoscenza del cluster, il sistema genera un messaggio di errore quando tale applicazione tenta di utilizzare le risorse del cluster.
- Quando si visualizzano le informazioni sullo stato per il gestore code, è possibile che si preferisca non visualizzare le informazioni in scadenza sui restanti membri del cluster.

Questa attività utilizza il cluster INVENTORY come esempio. Il gestore code LONDON è stato rimosso dal cluster INVENTORY come descritto in [“Rimozione di un gestore code da un cluster: procedura ottimale”](#) a pagina 375. Per eliminare la conoscenza dei restanti membri del cluster, immettere il seguente comando sul gestore code LONDON .

Procedura

1. Rimuovere tutta la memoria degli altri gestori code nel cluster da questo gestore code:

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. Monitorare il gestore code fino a quando tutte le risorse cluster non sono più disponibili:

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)  
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)  
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

Concetti correlati

[Cluster](#)

[Componenti cluster](#)

Riferimenti correlati

[Confronto tra cluster e accodamento distribuito](#)

Gestione di un gestore code

Sospendere e riprendere un gestore code da un cluster per eseguire la manutenzione.

Informazioni su questa attività

Di tanto in tanto, potrebbe essere necessario eseguire la manutenzione su un gestore code che fa parte di un cluster. Ad esempio, potrebbe essere necessario eseguire backup dei dati nelle relative code o applicare correzioni al software. Se il gestore code ospita delle code, le sue attività devono essere sospese. Una volta completata la manutenzione, è possibile riprendere le attività.

Procedura

1. Sospendere un gestore code immettendo il comando `SUSPEND QMGR runmqsc` :

```
SUSPEND QMGR CLUSTER(SALES)
```

Il comando `SUSPEND runmqsc` notifica ai gestori code nel cluster SALES che questo gestore code è stato sospeso.

Lo scopo del comando `SUSPEND QMGR` è solo quello di consigliare agli altri gestori code di evitare l'invio di messaggi a questo gestore code, se possibile. Ciò non significa che il gestore code sia disabilitato. Alcuni messaggi che devono essere gestiti da questo gestore code vengono ancora inviati ad esso, ad esempio quando questo gestore code è l'unico host di una coda cluster.

Mentre il gestore code è sospeso, le routine di gestione del carico di lavoro evitano di inviarle messaggi. I messaggi che devono essere gestiti da tale gestore code includono i messaggi inviati dal gestore code locale.

IBM MQ utilizza un algoritmo di bilanciamento del carico di lavoro per determinare quali destinazioni sono adatte, piuttosto che selezionare il gestore code locale quando possibile.

- a) Applicare la sospensione di un gestore code utilizzando l'opzione `FORCE` sul comando `SUSPEND QMGR` :

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

`MODE(FORCE)` arresta in modo forzato tutti i canali in entrata da altri gestori code nel cluster. Se non si specifica `MODE(FORCE)`, si applica il valore predefinito `MODE(QUIESCE)`.


2. Eseguire tutte le attività di manutenzione necessarie.
3. Riprendere il gestore code immettendo il comando `RESUME QMGR` **`runmqsc`** :

```
RESUME QMGR CLUSTER(SALES)
```


Risultati

Il comando di `RESUME` **`runmqsc`** notifica ai repository completi che il gestore code è nuovamente disponibile. I gestori code del repository completo diffondono queste informazioni ad altri gestori code che hanno richiesto aggiornamenti alle informazioni relative a questo gestore code.

Manutenzione della coda di trasmissione del cluster

Fare ogni sforzo per mantenere disponibili le code di trasmissione del cluster. Sono essenziali per le prestazioni dei cluster.  Su z/OS, impostare `INDXTYPE` di una coda di trasmissione cluster su `CORRELID`.

Prima di iniziare

- Assicurarsi che la coda di trasmissione del cluster non diventi piena.
- Fare attenzione a non immettere un comando `ALTER` **`runmqsc`** per impostarlo come disabilitato o disabilitato accidentalmente.
- Assicurarsi che il supporto della coda di trasmissione del cluster sia memorizzato su  (ad esempio, z/OS serie di pagine) non diventi pieno.

Informazioni su questa attività



La seguente procedura è applicabile solo a z/OS.

Procedura

Impostare `INDXTYPE` della coda di trasmissione cluster su `CORRELID`

Aggiornamento di un gestore code cluster

È possibile rimuovere i canali definiti automaticamente e gli oggetti cluster definiti automaticamente dal repository locale utilizzando il comando `REFRESH CLUSTER`. Nessun messaggio viene perso.

Prima di iniziare

Potrebbe essere richiesto di utilizzare il comando dal centro di supporto IBM. Non utilizzare il comando senza un'attenta considerazione. Ad esempio, per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può essere disruttivo per il cluster mentre è in corso e di nuovo a intervalli di 27 giorni quando gli oggetti cluster inviano automaticamente gli aggiornamenti di stato a tutti i gestori code interessati. Consultare [Cluster: utilizzo delle procedure ottimali di REFRESH CLUSTER](#).

Informazioni su questa attività

Un gestore code può avviare nuovamente un cluster. In circostanze normali, non è necessario utilizzare il comando `REFRESH CLUSTER`.

Procedura

Immettere il comando `REFRESH CLUSTER MQSC` da un gestore code per rimuovere il gestore code del cluster definito automaticamente e gli oggetti coda dal repository locale.

Il comando rimuove solo gli oggetti che fanno riferimento ad altri gestori code, non rimuove gli oggetti relativi al gestore code locale. Il comando rimuove anche i canali definiti automaticamente. Rimuove i canali che non hanno messaggi nella coda di trasmissione del cluster e che non sono collegati a un gestore code del repository completo.

Risultati

In effetti, il comando `REFRESH CLUSTER` consente a un gestore code di essere avviato a freddo rispetto al contenuto del repository completo. IBM MQ garantisce che non si perda alcun dato dalle code.

Informazioni correlate

[Cluster: utilizzo delle procedure consigliate per REFRESH CLUSTER](#)

Ripristino di un gestore code cluster

Aggiornare le informazioni del cluster su un gestore code utilizzando il comando `REFRESH CLUSTER runmqsc`. Seguire questa procedura dopo aver ripristinato un gestore code da un backup con riferimento temporale.

Prima di iniziare

È stato ripristinato un gestore code cluster da un backup point-in-time.

Informazioni su questa attività

Per recuperare un gestore code in un cluster, ripristinare il gestore code e aggiornare le relative informazioni utilizzando il comando `REFRESH CLUSTER runmqsc`.

Nota: Per i cluster di grandi dimensioni, l'utilizzo del comando **REFRESH CLUSTER** può danneggiare il cluster mentre è in esecuzione e, di nuovo, a intervalli di 27 giorni, quando gli oggetti del cluster inviano automaticamente aggiornamenti sullo stato a tutti i gestori code interessati. Consultare [Refreshing in a large cluster can affect performance and availability of the cluster](#).

Procedura

Immettere il comando `REFRESH CLUSTER` sul gestore code ripristinato per tutti i cluster a cui partecipa il gestore code.

Operazioni successive

Non è necessario immettere il comando REFRESH CLUSTER su un altro gestore code.

Concetti correlati

[Cluster: utilizzo delle procedure consigliate per REFRESH CLUSTER](#)

Configurazione dei canali cluster per disponibilità

Seguire le procedure di configurazione ottimali per mantenere i canali cluster in esecuzione senza problemi in caso di arresti di rete intermittenti.

Prima di iniziare

I cluster ti sollevano dalla necessità di definire i canali, ma devi comunque mantenerli. La stessa tecnologia di canale viene utilizzata per la comunicazione tra i gestori code in un cluster come viene utilizzata nell'accodamento distribuito. Per comprendere i canali cluster, è necessario avere dimestichezza con questioni quali:

- Funzionamento dei canali
- Come trovare il loro stato
- Come utilizzare le uscite canale

Informazioni su questa attività

Si consiglia di prestare particolare attenzione ai seguenti punti:

Procedura

Considerare i seguenti punti quando si configurano i canali cluster

- Scegliere i valori per HBINT o KAINTE sui canali mittenti del cluster e i canali riceventi del cluster che non caricano la rete con molti flussi heartbeat o keep alive. Un intervallo inferiore a circa 10 secondi fornisce falsi errori, se la rete a volte rallenta e introduce ritardi di questa lunghezza.
- Impostare il valore BATCHHB per ridurre la finestra per la causa di un messaggio di cui è stato eseguito il marooned poiché è in dubbio su un canale non riuscito. Un batch in dubbio su un canale non riuscito è più probabile che si verifichi se il batch viene fornito più a lungo da riempire. Se il traffico di messaggi lungo il canale è sporadico con lunghi periodi di tempo tra le interruzioni di messaggi, è più probabile che un batch non sia riuscito.
- Si verifica un problema se l'estremità mittente del cluster di un canale ha esito negativo e quindi tenta di riavviare prima che l'heartbeat o il keep alive abbia rilevato l'errore. Il riavvio del mittente del canale viene rifiutato se l'estremità ricevente del cluster del canale è rimasta attiva. Per evitare l'errore, fare in modo che il canale ricevente del cluster venga terminato e riavviato quando un canale mittente del cluster tenta il riavvio.

Attivo IBM MQ for z/OS

Controllare il problema dell'estremità ricevente del canale rimanente attivo utilizzando i parametri **ADOPTMCA** e **ADOPTCHK** su **ALTER QMGR**.

Attivo Multiplatforme

Controllare il problema dell'estremità ricevente del cluster del canale rimasto attivo utilizzando gli attributi **AdoptNewMCA**, **AdoptNewMCATimeout** e **AdoptNewMCACheck** nel file [qm.ini](#) o nel registro Windows .

Esempio

Consultare [“Impostazioni suggerite” a pagina 244](#) per esempi su come implementare queste impostazioni su tutte le piattaforme.

Verifica che i comandi asincroni per le reti distribuite siano terminati

Molti comandi sono asincroni quando utilizzati in una rete distribuita. A seconda del comando e dello stato della rete quando viene emesso, il completamento può richiedere una quantità di tempo significativa. Il gestore code non emette un messaggio al completamento, quindi sono necessari altri modi per controllare che il comando sia terminato.

Informazioni su questa attività

È probabile che quasi tutte le modifiche di configurazione apportate a un cluster vengano completate in modo asincrono. Ciò è dovuto ai cicli di gestione e aggiornamento interni che operano all'interno dei cluster. Per le gerarchie di pubblicazione / sottoscrizione, è probabile che qualsiasi modifica di configurazione che influisca sulle sottoscrizioni venga completata in modo asincrono. Questo non è sempre ovvio dal nome del comando.

I seguenti comandi MQSC potrebbero essere tutti completati in modo asincrono. Ognuno di questi comandi ha un equivalente PCF e la maggior parte è disponibile anche da IBM MQ Explorer . Quando vengono eseguiti su una rete di piccole dimensioni senza alcun carico di lavoro, questi comandi generalmente vengono completati in pochi secondi. Tuttavia, questo non vale per le reti più grandi e più affollate. Inoltre, il comando **REFRESH CLUSTER** potrebbe richiedere molto più tempo, in particolare quando viene emesso su più gestori code contemporaneamente.

Per avere la certezza che questi comandi siano terminati, verificare che gli oggetti previsti esistano sui gestori code remoti.

Procedura

- [ALTER DRG](#)

Per il comando `ALTER QMGR PARENT` , utilizzare `DISPLAY PUBSUB TYPE(PARENT) ALL` per tracciare lo stato della relazione principale richiesta.

Per i comandi `ALTER QMGR REPOS` e `ALTER QMGR REPOSNL` , utilizzare `DISPLAY CLUSQMGR QMTYPE` per confermare il completamento.

- [DEFINE CHANNEL](#), [ALTER CHANNEL](#)e [DELETE CHANNEL](#)

Per tutti i parametri elencati nella tabella [Parametri ALTER CHANNEL](#), utilizzare il comando `DISPLAY CLUSQMGR` per monitorare quando le modifiche sono state propagate al cluster.

- [DEFINE NAMELIST](#), [ALTER NAMELIST](#)e [DELETE NAMELIST](#).

Se si utilizza un **NAMELIST** sull'attributo **CLUSNL** di un oggetto **QMGR** , una coda o un canale cluster potrebbero influire su tale oggetto. Monitorare come appropriato per l'oggetto interessato.

Le modifiche a `SYSTEM.QPUBSUB.QUEUE.NAMELIST` potrebbero influire sulla creazione o l'annullamento delle sottoscrizioni proxy in una gerarchia di pubblicazione / sottoscrizione. Utilizzare il comando `DISPLAY SUB SUBTYPE(PROXY)` per monitorarlo.

- [DEFINE queues](#), [ALTER queue](#)e [DELETE queues](#).

Per tutti i parametri elencati nella tabella [Parametri che possono essere restituiti dal comando DISPLAY QUEUE](#), utilizzare il comando `DISPLAY QCLUSTER` per monitorare quando le modifiche sono state propagate al cluster.

- [DEFINE SUB](#)e [DELETE SUB](#)

Quando si definisce la prima sottoscrizione su una stringa argomento, è possibile creare sottoscrizioni proxy in una gerarchia di pubblicazione / sottoscrizione o in un cluster di pubblicazione / sottoscrizione. Allo stesso modo, quando si elimina l'ultima sottoscrizione su una stringa di argomenti, è possibile annullare le sottoscrizioni proxy in una gerarchia di pubblicazione / sottoscrizione o in un cluster di pubblicazione / sottoscrizione.

Per verificare che un comando che definisce o elimina una sottoscrizione sia terminato, verificare se la sottoscrizione proxy prevista esiste o meno su altri gestori code nella rete distribuita. Se si sta utilizzando l' *instradamento diretto* in un cluster, verificare che la sottoscrizione proxy prevista esista

sugli altri repository parziali nel cluster. Se si sta utilizzando l' *instradamento dell'host argomento* in un cluster, verificare che la sottoscrizione proxy prevista esista sugli host argomento corrispondenti. Utilizzare il seguente comando MQSC:

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

Utilizzare lo stesso controllo per le seguenti chiamate MQI di sottoscrizione e annullamento sottoscrizione equivalenti, quando vengono emesse in un cluster o gerarchia:

- Sottoscrivere utilizzando [MQSUB](#).
- Annullare la sottoscrizione utilizzando [MQCLOSE](#) con MQCO_REMOVE_SUB.
- [DEFINE TOPIC](#), [ALTER TOPIC](#) e [DELETE TOPIC](#)

Per verificare che un comando che definisce, modifica o elimina un argomento del cluster sia terminato, visualizzare l'argomento negli altri repository parziali nel cluster (se si utilizza l' *instradamento diretto*) o su altri host argomento (se si utilizza l' *instradamento host argomento*).

Per tutti i parametri elencati nella tabella [Parametri che possono essere restituiti dal comando DISPLAY TOPIC](#), utilizzare il comando `DISPLAY TCLUSTER` per monitorare quando le modifiche sono state propagate al cluster.

Nota:

- Il parametro **CLUSTER** può influenzare la creazione o l'annullamento delle sottoscrizioni proxy in un cluster di pubblicazione / sottoscrizione.
- I parametri **PROXYSUB** e **SUBSCOPE** possono influenzare la creazione o l'annullamento delle sottoscrizioni proxy in una gerarchia di pubblicazione / sottoscrizione o in un cluster di pubblicazione / sottoscrizione.
- Utilizzare il comando `DISPLAY SUB SUBTYPE(PROXYSUB)` per monitorarlo.
- [Aggiornamento cluster](#)

Se si sta eseguendo il comando **REFRESH CLUSTER** , eseguire il polling della profondità della coda comandi del cluster. Attendere che raggiunga lo zero, e rimanere a zero, prima di ricercare gli oggetti.

1. Utilizzare il comando MQSC riportato di seguito per controllare che la profondità della coda comandi del cluster sia zero.

```
DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
```

2. Ripetere il controllo fino a quando la profondità della coda non raggiunge lo zero e rimane a zero nel successivo controllo.

Il comando **REFRESH CLUSTER** rimuove e ricrea oggetti e, in configurazioni di grandi dimensioni, può richiedere molto tempo per il completamento. Consultare [Considerazioni su REFRESH CLUSTER per i cluster di pubblicazione/sottoscrizione](#).

- [AGGIORNA QMGR TYPE \(PROXYSUB\)](#)

Per verificare che il comando **REFRESH QMGR TYPE (PROXYSUB)** sia terminato, verificare che le sottoscrizioni proxy siano state corrette su altri gestori code nella rete distribuita. Se si sta utilizzando l' *instradamento diretto* in un cluster, verificare che le sottoscrizioni proxy siano state corrette sugli altri repository parziali nel cluster. Se si sta utilizzando l' *instradamento host argomento* in un cluster, verificare che le sottoscrizioni proxy previste siano state corrette sugli host argomento corrispondenti. Utilizzare il seguente comando MQSC:

```
DISPLAY SUB(*) SUBTYPE(PROXYSUB)
```

- [Reimposta cluster](#)

Per controllare che il comando **RESET CLUSTER** è stato completato, utilizzare `DISPLAY CLUSQMGR`.

- [RESET QMGR TYPE \(PUBSUB\)](#)

Per controllare che il comando **RESET QMGR** è stato completato, utilizzare DISPLAY PUBSUB TYPE (PARENT | CHILD).

Nota: Il comando **RESET QMGR** potrebbe causare l'annullamento delle sottoscrizioni proxy in una gerarchia di pubblicazione / sottoscrizione o in un cluster di pubblicazione / sottoscrizione. Utilizzare il comando DISPLAY SUB SUBTYPE (PROXYSUB) per monitorarlo.

- Si potrebbe anche voler monitorare altre code di sistema che, man mano che i comandi vengono completati, tendono ad una profondità di coda pari a zero.

Ad esempio, si potrebbe voler monitorare la coda SYSTEM.INTER.QMGR.CONTROL e la coda SYSTEM.INTER.QMGR.FANREQ. Vedi [Monitoraggio del traffico di sottoscrizioni proxy nei cluster e Bilanciamento dei produttori e dei consumatori nelle reti di pubblicazione / sottoscrizione](#).

Operazioni successive

Se questi controlli non confermano che un comando asincrono è terminato, è possibile che si sia verificato un errore. Per esaminare, controllare prima il log del gestore code su cui è stato emesso il comando, quindi (per un cluster) controllare i log del repository completo del cluster.

Riferimenti correlati

 [Comportamento asincrono dei comandi CLUSTER su z/OS](#)

Instradamento dei messaggi verso e dai cluster

Utilizzare gli alias di coda, gli alias di gestore code e le definizioni di code remote per connettere i cluster a gestori code esterni e altri cluster.

Per i dettagli sull'instradamento dei messaggi verso e dai cluster, consultare i topic secondari riportati di seguito:

Concetti correlati

[Cluster](#)

[Componenti di un cluster](#)

[“Cluster e alias del gestore code” a pagina 400](#)

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

[“Alias coda e cluster” a pagina 404](#)

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

[“Cluster e alias della coda di risposta” a pagina 403](#)

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Attività correlate

[“Configurazione di un cluster di gestore code” a pagina 303](#)

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. È possibile definire componenti cluster e creare e gestire cluster.

[“Configurazione di un nuovo cluster” a pagina 318](#)

Seguire queste istruzioni per configurare il cluster di esempio. Istruzioni separate descrivono l'impostazione del cluster su TCP/IP, LU 6.2e con una o più code di trasmissione. Verificare il funzionamento del cluster inviando un messaggio da un gestore code all'altro.

Riferimenti correlati

[Confronto tra cluster e accodamento distribuito](#)

Configurazione della richiesta/risposta a un cluster

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

Prima di iniziare

La Figura 53 a pagina 388 mostra un gestore code denominato QM3 esterno al cluster denominato DEMO. QM3 potrebbe essere un gestore code su un prodotto IBM MQ che non supporta i cluster. QM3 ospita una coda denominata Q3, definita come segue:

```
DEFINE QLOCAL(Q3)
```

All'interno del cluster sono presenti due gestori code denominati QM1 e QM2. QM2 ospita una coda cluster denominata Q2, definita come segue:

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

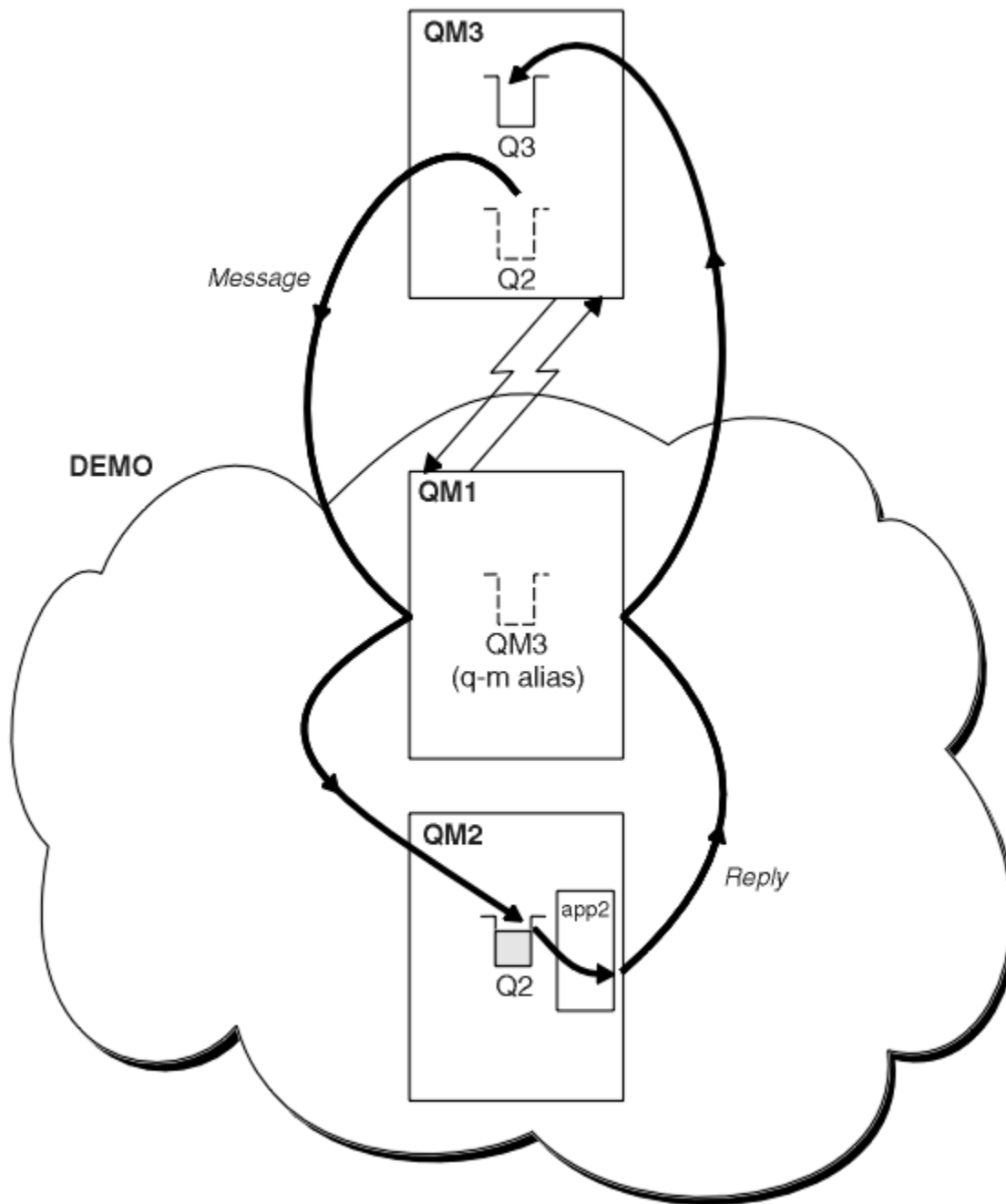


Figura 53. Inserimento da un gestore code esterno al cluster

Informazioni su questa attività

Seguire il consiglio nella procedura per impostare il percorso per i messaggi di richiesta e risposta.

Procedura

1. Inviare il messaggio di richiesta al cluster.

Considerare come il gestore code esterno al cluster inserisce un messaggio nella coda Q2 in QM2, all'interno del cluster. Un gestore code esterno al cluster deve avere una definizione QREMOTE per ogni coda nel cluster in cui inserisce i messaggi.

- a) Definire una coda remota per Q2 su QM3.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
```

Poiché QM3 non fa parte di un cluster, deve comunicare utilizzando tecniche di accodamento distribuite. Pertanto, deve avere anche un canale mittente e una coda di trasmissione a QM1. QM1 necessita di un canale ricevente corrispondente. I canali e le code di trasmissione non vengono visualizzati esplicitamente in [Figura 53 a pagina 388](#).

Nell'esempio, un'applicazione in QM3 emette una chiamata MQPUT per inserire un messaggio in Q2. La definizione QREMOTE fa in modo che il messaggio venga instradato a Q2 in QM2 utilizzando il canale mittente che sta ricevendo i messaggi dalla coda di trasmissione QM1 .

2. Ricevere il messaggio di risposta dal cluster.

Utilizzare un alias del gestore code per creare un percorso di ritorno per le risposte a un gestore code esterno al cluster. Il gateway, QM1, annuncia un alias del gestore code per il gestore code esterno al cluster, QM3. Annuncia QM3 ai gestori code all'interno del cluster aggiungendo l'attributo cluster a una definizione di alias del gestore code per QM3. Una definizione alias del gestore code è come una definizione di coda remota, ma con un RNAME vuoto.

a) Definire un alias del gestore code per QM3 su QM1.

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

È necessario considerare la scelta del nome della coda di trasmissione utilizzata per inoltrare le risposte da QM1 a QM3. Implicito nella definizione QREMOTE , per omissione dell'attributo XMITQ , il nome della coda di trasmissione è QM3. Ma QM3 è lo stesso nome che ci aspettiamo di pubblicizzare al resto del cluster utilizzando l'alias del gestore code. IBM MQ non consente di assegnare lo stesso nome alla coda di trasmissione e all'alias del gestore code. Una soluzione è quella di creare una coda di trasmissione per inoltrare i messaggi a QM3 con un nome diverso all'alias del gestore code.

b) Fornire il nome della coda di trasmissione nella definizione QREMOTE .

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
```

Il nuovo alias del gestore code accoppia la nuova coda di trasmissione denominata QM3 .XMIT con l'alias del gestore code QM3 . Si tratta di una soluzione semplice e corretta, ma non del tutto soddisfacente. È stata interrotta la convenzione di denominazione per le code di trasmissione per cui viene assegnato loro lo stesso nome del gestore code di destinazione. Esistono soluzioni alternative che preservano la convenzione di denominazione della coda di trasmissione?

Il problema si verifica perché il richiedente utilizza il valore predefinito QM3 come nome del gestore code di risposta nel messaggio di richiesta inviato da QM3. Il server su QM2 utilizza il QM3 nome del gestore code reply - to QM3 nelle sue risposte. La soluzione richiedeva QM1 per pubblicizzare QM3 come alias del gestore code a cui restituire i messaggi di risposta e impediva a QM1 di utilizzare QM3 come nome della coda di trasmissione.

Invece di fornire per impostazione predefinita QM3 come nome del gestore code di risposta, le applicazioni su QM3 devono passare un alias del gestore code di risposta a QM1 per i messaggi di risposta. Il gestore code del gateway QM1 pubblicizza l'alias del gestore code per le risposte a QM3 piuttosto che QM3 , evitando il conflitto con il nome della coda di trasmissione.

c) Definire un alias del gestore code per QM3 su QM1.

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

Sono richieste due modifiche ai comandi di configurazione.

- i) QREMOTE at QM1 ora annuncia il nostro alias del gestore code QM3 . ALIAS al resto del cluster, accoppiandolo al nome del gestore code reale QM3. QM3 è nuovamente il nome della coda di trasmissione a cui inviare le code di risposte QM3

- ii) L'applicazione client deve fornire QM3 . ALIAS come nome del gestore code di risposta quando crea il messaggio di richiesta. È possibile fornire QM3 . ALIAS all'applicazione client in uno dei seguenti due modi.
- Codice QM3 . ALIAS nel campo del nome del gestore code di risposta creato da MQPUT in MQMD. È necessario farlo in questo modo se si utilizza una coda dinamica per le risposte.
 - Utilizzare un alias della coda di risposta, Q3 . ALIAS, piuttosto che una coda di risposta quando si fornisce il nome della coda di risposta.

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

Operazioni successive

Nota: Non è possibile dimostrare l'utilizzo di alias della coda di risposta con **AMQSREQ0**. Apre la coda di risposta utilizzando il nome della coda fornito nel parametro 3 o la coda modello SYSTEM . SAMPLE . REPLY predefinita. È necessario modificare l'esempio fornendo un altro parametro contenente l'alias della coda di risposta per denominare l'alias del gestore code di risposta per MQPUT.

Concetti correlati

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

Attività correlate

Configurazione della richiesta/risposta da un cluster

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Configurazione dei percorsi dei messaggi tra cluster

Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

“Nascondere il nome di un gestore code di destinazione cluster” a pagina 390

Instradare un messaggio a una coda cluster definita su qualsiasi gestore code in un cluster senza denominare il gestore code.

Nascondere il nome di un gestore code di destinazione cluster

Instradare un messaggio a una coda cluster definita su qualsiasi gestore code in un cluster senza denominare il gestore code.

Prima di iniziare

- Evitare di rivelare i nomi dei gestori code interni al cluster ai gestori code esterni al cluster.

- La risoluzione dei riferimenti al gestore code che ospita una coda all'interno del cluster rimuove la flessibilità per eseguire il bilanciamento del carico di lavoro.
 - Inoltre, rende difficile modificare un gestore code che ospita una coda nel cluster.
 - L'alternativa consiste nel sostituire RQMNAME con un alias del gestore code fornito dall'amministratore del cluster.
 - [“Nascondere il nome di un gestore code di destinazione cluster”](#) a pagina 390 descrive l'utilizzo di un alias del gestore code per separare un gestore code esterno a un cluster dalla gestione dei gestori code all'interno di un cluster.
- Tuttavia, il modo consigliato per denominare le code di trasmissione consiste nel fornire loro il nome del gestore code di destinazione. Il nome della coda di trasmissione rivela il nome di un gestore code nel cluster. Devi scegliere quale regola seguire. È possibile scegliere di denominare la coda di trasmissione utilizzando il nome del gestore code o il nome del cluster:

Denominare la coda di trasmissione utilizzando il nome del gestore code del gateway

La divulgazione del nome del gestore code del gateway ai gestori code all'esterno di un cluster è un'eccezione ragionevole alla regola di nascondere i nomi dei gestori code del cluster.

Denominare la coda di trasmissione utilizzando il nome del cluster

Se non si sta seguendo la convenzione di denominazione delle code di trasmissione con il nome del gestore code di destinazione, utilizzare il nome cluster.

Informazioni su questa attività

Modificare l'attività [“Configurazione della richiesta/risposta a un cluster”](#) a pagina 387, per nascondere il nome del gestore code di destinazione all'interno del cluster.

Procedura

Nell'esempio, consultare [Figura 54 a pagina 392](#), definire un alias del gestore code sul gestore code del gateway QM1 denominato DEMO:

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

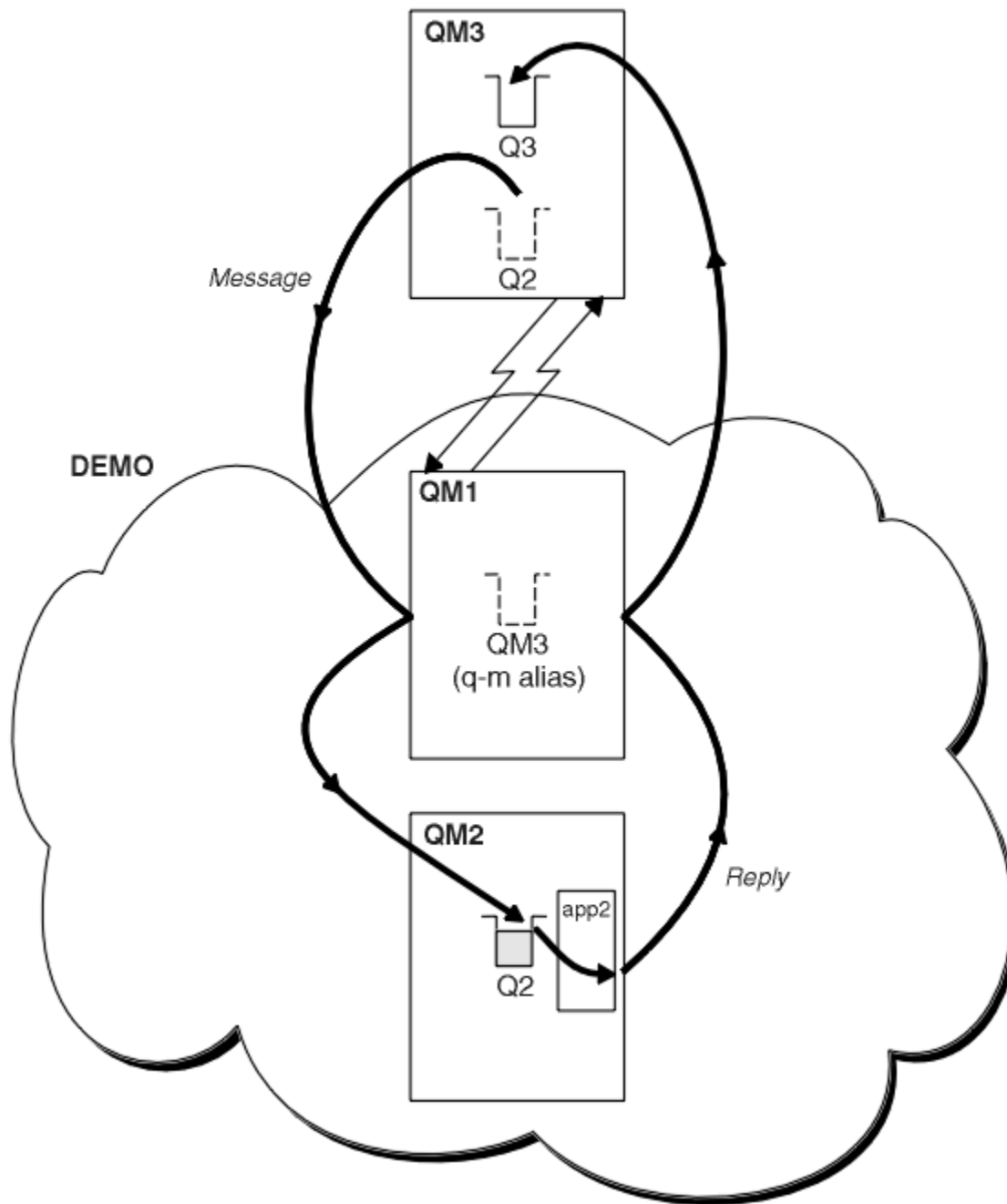


Figura 54. Inserimento da un gestore code esterno al cluster

La definizione QREMOTE su QM1 rende l'alias del gestore code DEMO noto al gestore code gateway. QM3, il gestore code esterno al cluster, può utilizzare l'alias del gestore code DEMO per inviare messaggi alle code del cluster su DEMO, invece di dover utilizzare un nome gestore code effettivo.

Se si adotta la convenzione di utilizzare il nome del cluster per denominare la coda di trasmissione che si connette a un cluster, la definizione della coda remota per Q2 diventa:

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO) XMIT(DEMO)
```


Risultati

I messaggi destinati a Q2 su DEMO vengono inseriti nella coda di trasmissione DEMO . Dalla coda di trasmissione vengono trasferiti dal canale mittente al gestore code del gateway, QM1. Il gestore code del gateway instrada i messaggi a qualsiasi gestore code nel cluster che ospita la coda del cluster Q2.

Configurazione della richiesta/risposta da un cluster

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

Prima di iniziare

Figura 55 a pagina 394 mostra un gestore code, QM2, all'interno del cluster DEMO. Invia una richiesta a una coda, Q3, che si trova sul gestore code esterno al cluster. Le risposte vengono restituite a Q2 in QM2 all'interno del cluster.

Per comunicare con il gestore code esterno al cluster, uno o più gestori code all'interno del cluster fungono da gateway. Un gestore code gateway dispone di un percorso di comunicazione per i gestori code esterni al cluster. Nell'esempio, QM1 è il gateway.

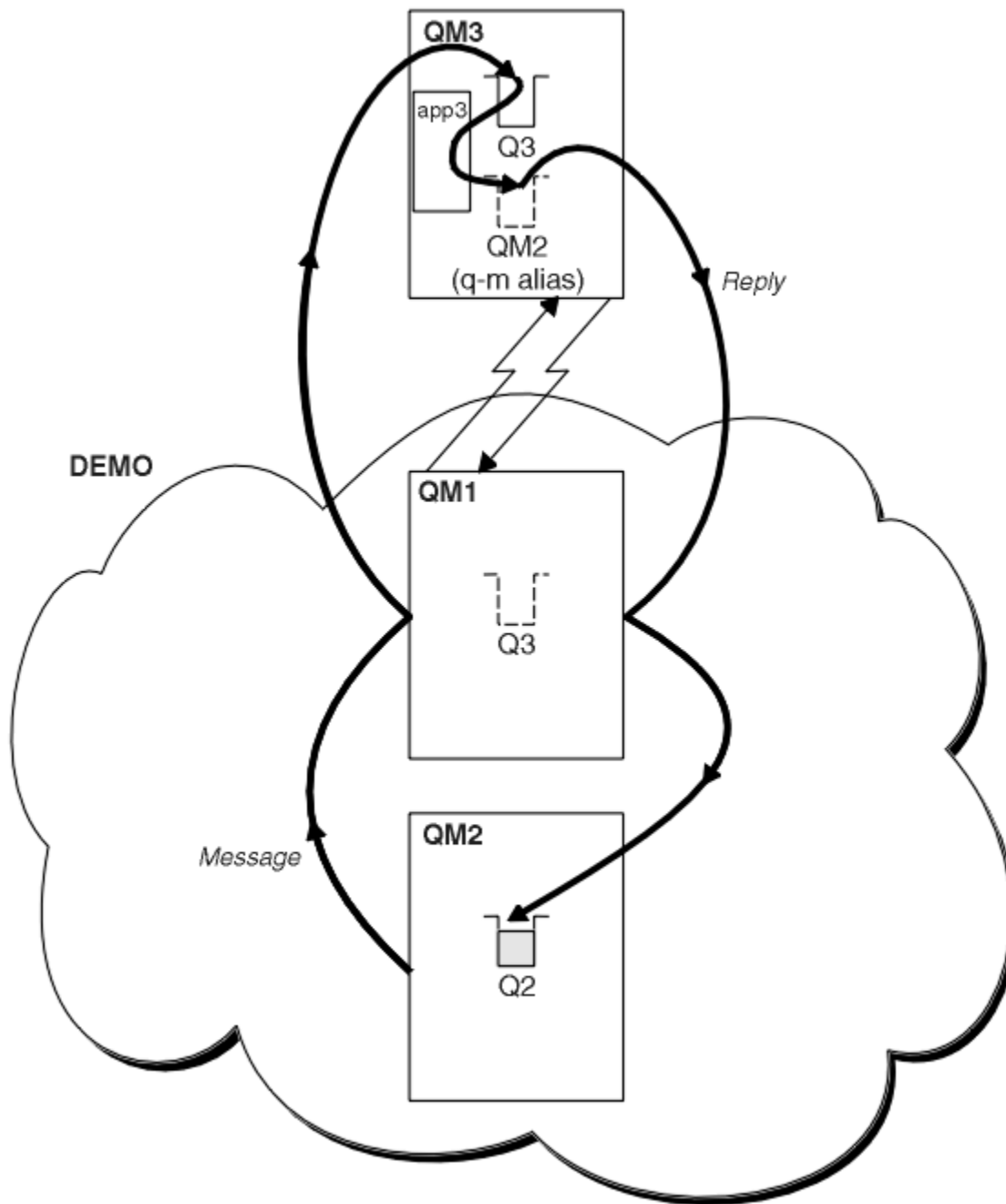


Figura 55. Inserimento in un gestore code all'esterno del cluster

Informazioni su questa attività

Seguire le istruzioni per impostare il percorso per i messaggi di richiesta e risposta

Procedura

1. Invia il messaggio di richiesta dal cluster.

Considerare il modo in cui il gestore code, QM2, che si trova nel cluster inserisce un messaggio nella coda Q3 in QM3, che si trova all'esterno del cluster.

- a) Creare una definizione QREMOTE su QM1 che annunci la coda remota Q3 al cluster

```
DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
```

Dispone inoltre di un canale mittente e di una coda di trasmissione al gestore code esterno al cluster. QM3 ha un canale ricevente corrispondente. I canali non vengono visualizzati in [Figura 55 a pagina 394](#).

Un'applicazione su QM2 emette una chiamata MQPUT che specifica la coda di destinazione e la coda a cui devono essere inviate le risposte. La coda di destinazione è Q3 e la coda di risposta è Q2.

Il messaggio viene inviato a QM1, che utilizza la definizione della coda remota per risolvere il nome della coda in Q3 at QM3.

2. Ricevere il messaggio di risposta dal gestore code esterno al cluster.

Un gestore code all'esterno del cluster deve avere un alias del gestore code per ogni gestore code nel cluster a cui invia un messaggio. L'alias del gestore code deve anche specificare il nome della coda di trasmissione al gestore code gateway. In questo esempio, QM3 ha bisogno di una definizione alias del gestore code per QM2:

a) Creare un alias del gestore code QM2 su QM3

```
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
```

QM3 ha anche bisogno di un canale mittente e di una coda di trasmissione per QM1 e QM1 ha bisogno di un canale ricevente corrispondente.

L'applicazione, **app3**, su QM3 può quindi inviare risposte a QM2, emettendo una chiamata MQPUT e specificando il nome della coda, Q2 e il nome del gestore code, QM2.

Operazioni successive

È possibile definire più di un instradamento da un cluster.

Concetti correlati

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

Attività correlate

Configurazione della richiesta/risposta a un cluster

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Configurazione dei percorsi dei messaggi tra cluster

Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Prima di iniziare

Configurare l'esempio, come mostrato in [Figura 53 a pagina 388](#) in “Configurazione della richiesta/risposta a un cluster” a pagina 387.

Informazioni su questa attività

In questo scenario, il gestore code all'esterno del cluster, QM3 in [Figura 56 a pagina 397](#), invia richieste alla coda Q2. Q2 è ospitato su due gestori code, QM2 e QM4 all'interno del cluster DEMO. Entrambi i gestori code sono configurati con un'opzione di bind predefinita NOTFIXED per utilizzare il bilanciamento del carico di lavoro. Le richieste provenienti da QM3, il gestore code esterno al cluster, vengono inviate a un'istanza di Q2 tramite QM1.

QM3 non fa parte di un cluster e comunica utilizzando tecniche di accodamento distribuite. Deve avere un canale mittente e una coda di trasmissione per QM1. QM1 necessita di un canale ricevente corrispondente. I canali e le code di trasmissione non vengono visualizzati esplicitamente in [Figura 56 a pagina 397](#).

La procedura estende l'esempio in [Figura 53 a pagina 388](#) in “Configurazione della richiesta/risposta a un cluster” a pagina 387.

Procedura

1. Creare una definizione QREMOTE per Q2 su QM3.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
```

Creare una definizione QREMOTE per ogni coda nel cluster in cui QM3 inserisce i messaggi.

2. Creare un alias del gestore code Q3 su QM1.

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
```

Q3 non è un nome gestore code reale. È il nome di una definizione alias del gestore code nel cluster che equipara il nome alias del gestore code Q3 con vuoto, ' '.

3. Definire una coda locale denominata Q2 su ciascuno di QM2 e QM4.

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

4. QM1, il gestore code del gateway, non ha definizioni speciali.

Risultati

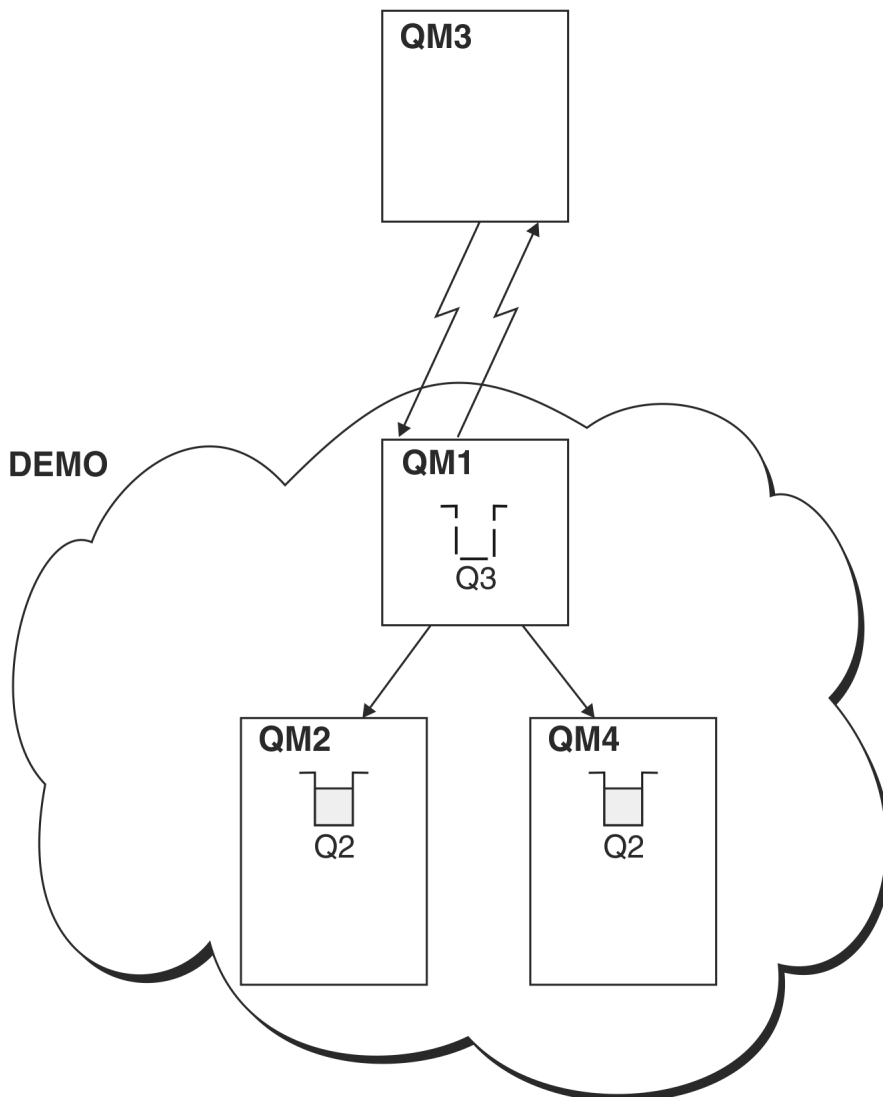


Figura 56. Inserimento da un gestore code esterno al cluster

Quando un'applicazione in QM3 emette una chiamata MQPUT per inserire un messaggio in Q2, la QREMOTE definizione in QM3 fa in modo che il messaggio venga instradato attraverso il gestore code gateway QM1. Quando QM1 riceve il messaggio, è consapevole che il messaggio è ancora destinato a una coda denominata Q2 ed esegue la risoluzione dei nomi. QM1 controlla le relative definizioni locali e non ne trova alcuna per Q2. QM1, quindi, controlla la configurazione del cluster e rileva due istanze di Q2 nel cluster DEMO. QM1 può ora utilizzare il bilanciamento del carico di lavoro per distribuire i messaggi tra le istanze di Q2 che risiedono su QM2 e QM4.

Concetti correlati

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

[Risoluzione nomi](#)

Attività correlate

[Configurazione della richiesta/risposta a un cluster](#)

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

[Configurazione della richiesta/risposta da un cluster](#)

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

[Configurazione dei percorsi dei messaggi tra cluster](#)

Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

Riferimenti correlati

[Risoluzione nome coda](#)

Configurazione dei percorsi dei messaggi tra cluster

Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

Informazioni su questa attività

Invece di raggruppare tutti i tuoi gestori code in un unico cluster di grandi dimensioni, puoi avere molti cluster più piccoli. Ogni cluster ha uno o più gestori code che fungono da bridge. Il vantaggio è che è possibile limitare la visibilità dei nomi di code e gestori code nei cluster. Vedi [Cluster sovrapposti](#). Utilizzare gli alias per modificare i nomi delle code e dei gestori code per evitare conflitti di nomi o per rispettare le convenzioni di denominazione locali.

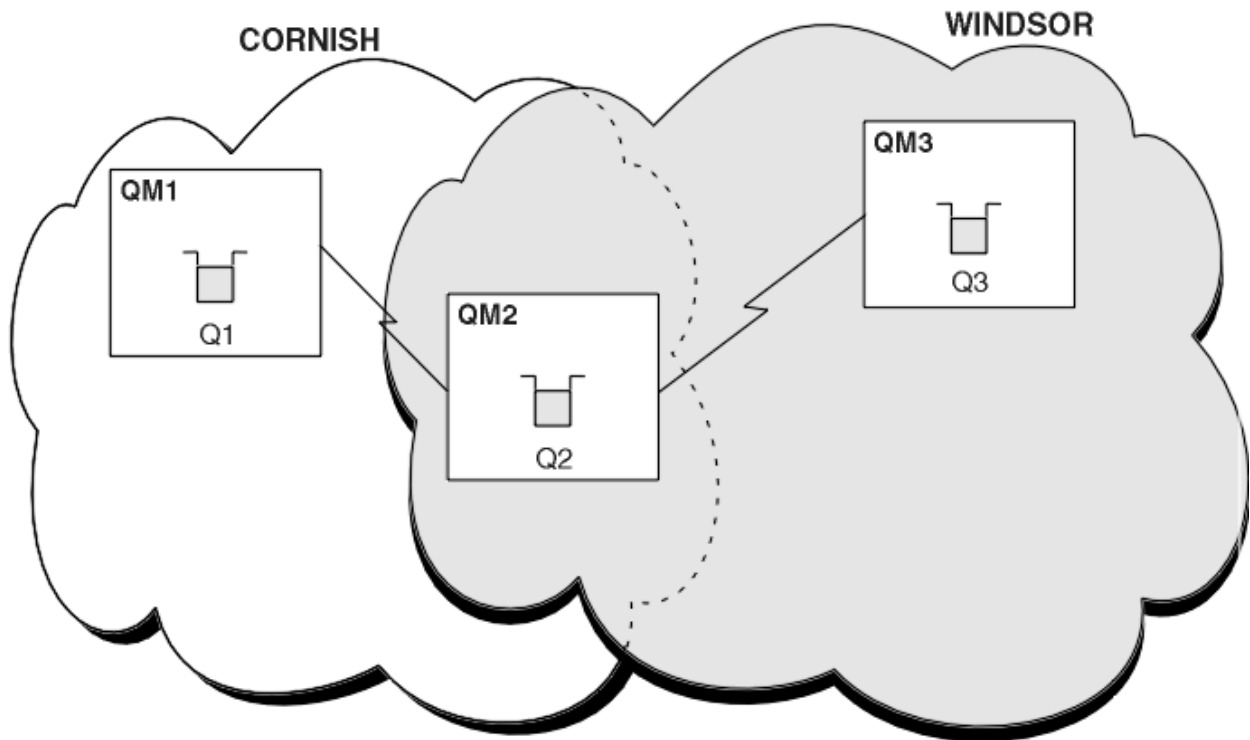


Figura 57. Collegamento tra cluster

Figura 57 a pagina 399 mostra due cluster con un bridge tra loro. Ci potrebbe essere più di un ponte.

Configurare i cluster utilizzando la seguente procedura:

Procedura

1. Definire una coda cluster, Q1 su QM1.

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. Definire una coda cluster, Q3 su QM3.

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. Creare un elenco nomi denominato CORNISHWINDSOR su QM2, contenente i nomi di entrambi i cluster.

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')
NAMES(CORNISH, WINDSOR)
```

4. Definire una coda cluster, Q2 su QM2

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

Operazioni successive

QM2 è un membro di entrambi i cluster ed è il ponte tra loro. Per ogni coda che si desidera rendere visibile attraverso il bridge, è necessaria una definizione QALIAS sul bridge. Ad esempio, in Figura 57 a pagina 399, su QM2, è necessario:

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

Utilizzando l'alias della coda, un'applicazione connessa a un gestore code in CORNISH, ad esempio QM1, può inserire un messaggio in Q3. Si riferisce a Q3 come MYQ3. Il messaggio viene instradato a Q3 in QM3.

Quando si apre una coda, è necessario impostare DEFBIND su NOTFIXED o QDEF. Se DEFBIND viene lasciato come valore predefinito, OPEN, il gestore code risolve la definizione dell'alias nel gestore code del bridge che lo ospita. Il bridge non inoltra il messaggio.

Per ogni gestore code che si desidera rendere visibile, è necessaria una definizione alias del gestore code. Ad esempio, su QM2 è necessario:

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

Un'applicazione connessa a qualsiasi gestore code in WINDSOR, ad esempio QM3, può inserire un messaggio in qualsiasi coda su QM1, denominando esplicitamente QM1 nella chiamata MQOPEN .

Concetti correlati

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

Attività correlate

Configurazione della richiesta/risposta a un cluster

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

Configurazione della richiesta/risposta da un cluster

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Gli alias dei gestori code, creati utilizzando una definizione di coda remota con un RNAMEvuoto, hanno cinque utilizzi:

Riassociazione del nome del gestore code durante l'invio di messaggi

Un alias del gestore code può essere utilizzato per riassociare il nome del gestore code specificato in una chiamata MQOPEN ad un altro gestore code. Può essere un gestore code cluster. Ad esempio, un gestore code potrebbe avere la definizione di alias del gestore code:

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

YORK può essere utilizzato come un alias per il gestore code denominato CLUSQM. Quando un'applicazione sul gestore code che ha creato questa definizione inserisce un messaggio nel gestore code YORK, il gestore code locale risolve il nome in CLUSQM. Se il gestore code locale non è

denominato CLUSQM, inserisce il messaggio nella coda di trasmissione del cluster da spostare in CLUSQM. Inoltre, modifica l'intestazione di trasmissione per dire CLUSQM invece che YORK.

Nota: La definizione viene applicata solo al gestore code che la crea. Per indicare l'alias all'intero cluster, è necessario aggiungere l'attributo CLUSTER alla definizione della coda remota. Quindi, i messaggi provenienti da altri gestori code destinati a YORK vengono inviati a CLUSQM.

Modifica o specifica della coda di trasmissione durante l'invio di messaggi

L'aliasing può essere utilizzato per unire un cluster a un sistema non cluster. Ad esempio, i gestori code nel cluster ITALY potrebbero comunicare con il gestore code denominato PALERMO, esterno al cluster. Per comunicare, uno dei gestori code nel cluster deve agire come gateway. Dal gestore code del gateway, immettere il comando:

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

Il comando è una definizione alias del gestore code. Definisce e pubblicizza ROME come un gestore code su cui i messaggi provenienti da qualsiasi gestore code nel cluster ITALY possono multi-hop per raggiungere la loro destinazione in PALERMO. I messaggi inseriti in una coda aperta con nome gestore code impostato su ROME vengono inviati al gestore code del gateway con la definizione di alias del gestore code. Una volta lì, i messaggi vengono inseriti sulla coda di trasmissione X e spostati dai canali non cluster al gestore code PALERMO.

La scelta del nome ROME in questo esempio non è significativa. I valori per QREMOTE e RQMNAME potrebbero coincidere.

Determinazione della destinazione durante la ricezione dei messaggi

Quando un gestore code riceve un messaggio, estrae il nome della coda di destinazione e il gestore code dall'intestazione di trasmissione. Cerca una definizione di alias del gestore code con lo stesso nome del gestore code nell'intestazione di trasmissione. Se ne trova uno, sostituisce RQMNAME dalla definizione dell'alias del gestore code per il nome del gestore code nell'intestazione di trasmissione.

Esistono due ragioni per utilizzare un alias del gestore code in questo modo:

- Per indirizzare i messaggi ad un altro gestore code
- Per modificare il nome del gestore code in modo che sia uguale al gestore code locale

Utilizzo degli alias del gestore code in un gestore code gateway per instradare i messaggi tra i gestori code in cluster differenti.

Un'applicazione può inviare un messaggio a una coda in un cluster differente utilizzando un alias del gestore code. La coda non deve essere una coda cluster. La coda è definita in un cluster. L'applicazione è connessa a un gestore code in un cluster differente. Un gestore code del gateway connette i due cluster. Se la coda non è definita come in cluster, per eseguire l'instradamento corretto, l'applicazione deve aprire la coda utilizzando il nome della coda e un nome alias del gestore code in cluster. Per un esempio di configurazione, vedere [“Creazione di due cluster sovrapposti con un gestore code del gateway” a pagina 352](#), da cui viene preso il flusso di messaggi di risposta illustrato nella figura 1.

Il diagramma mostra il percorso utilizzato dal messaggio di risposta per tornare a una coda dinamica temporanea, denominata RQ. L'applicazione server, connessa a QM3, apre la coda di risposta utilizzando il nome del gestore code QM2. Il nome del gestore code QM2 è definito come alias del gestore code in cluster su QM1. QM3 instrada il messaggio di risposta a QM1. QM1 instrada il messaggio a QM2.

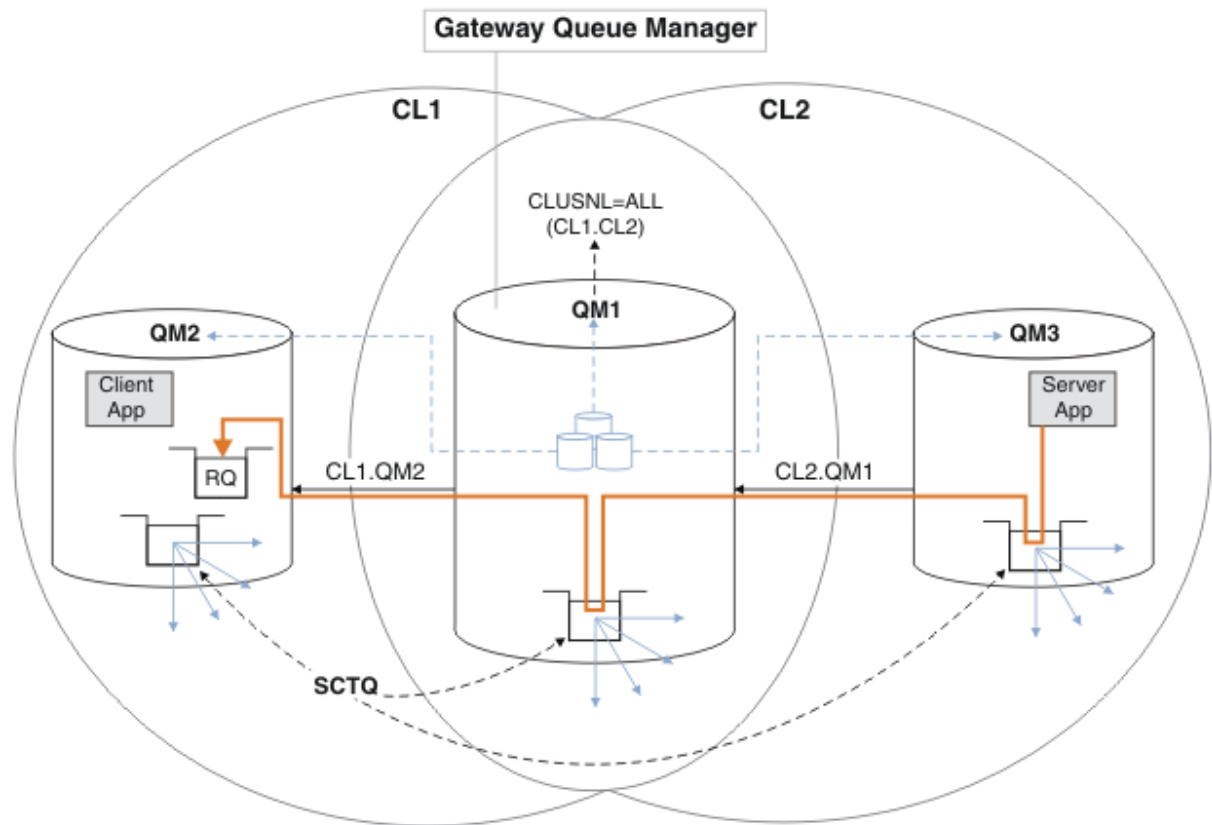


Figura 58. Utilizzo di un alias del gestore code per restituire il messaggio di risposta a un cluster differente

Il modo in cui funziona l'instradamento è il seguente. Ogni gestore code in ogni cluster ha una definizione di alias del gestore code su QM1. Gli alias sono raggruppati in tutti i cluster. Le frecce tratteggiate grigie da ciascuno degli alias a un gestore code mostrano che ogni alias del gestore code viene risolto in un gestore code reale in almeno uno di questi cluster. In questo caso, l'alias QM2 è raggruppato in cluster CL1 e CL2 e viene risolto nel gestore code reale QM2 in CL1. L'applicazione server crea il messaggio di risposta utilizzando il nome della coda di risposta RQ e il nome del gestore code di risposta QM2. Il messaggio viene instradato a QM1 perché la definizione alias del gestore code QM2 è definita su QM1 nel cluster CL2 e il gestore code QM2 non è nel cluster CL2. Poiché il messaggio non può essere inviato al gestore code di destinazione, viene inviato al gestore code che ha la definizione alias.

QM1 colloca il messaggio nella coda di trasmissione del cluster su QM1 per il trasferimento a QM2. QM1 instrada il messaggio a QM2 perché la definizione dell'alias del gestore code su QM1 per QM2 definisce QM2 come il gestore code di destinazione reale. La definizione non è circolare, poiché le definizioni alias possono fare riferimento solo a definizioni reali; l'alias non può puntare a se stesso. La definizione reale viene risolta da QM1, perché sia QM1 che QM2 si trovano nello stesso cluster, CL1. QM1 rileva le informazioni di collegamento per QM2 dal contenitore per CL1 e instrada il messaggio a QM2. Perché il messaggio venga reinstradato da QM1, l'applicazione server deve aver aperto la coda di risposta con l'opzione DEFBIND impostata su MQBND_BIND_NOT_FIXED. Se l'applicazione server ha aperto la coda di risposta con l'opzione MQBND_BIND_ON_OPEN, il messaggio non viene reinstradato e finisce su una coda di messaggi non recapitabili.

Utilizzo di un gestore code come gateway nel cluster per bilanciare il carico di lavoro dei messaggi provenienti dall'esterno del cluster.

Definire una coda denominata EDINBURGH su più di un gestore code nel cluster. Si desidera che il meccanismo di cluster bilanci il carico di lavoro per i messaggi che arrivano a tale coda dall'esterno del cluster.

Un gestore code esterno al cluster richiede una coda di trasmissione e un canale mittente per un gestore code nel cluster. Questa coda è denominata gestore code gateway. Per sfruttare il meccanismo di bilanciamento del workload predefinito, è necessario applicare una delle seguenti regole:

- Il gestore code del gateway non deve contenere un'istanza della coda EDINBURGH .
- Il gestore code del gateway specifica CLWLUSEQ (ANY) su ALTER QMGR.

Per un esempio di bilanciamento del carico di lavoro esterno a un cluster, consultare [“Configurazione del bilanciamento del workload dall'esterno di un cluster”](#) a pagina 395

Concetti correlati

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

Attività correlate

Configurazione della richiesta/risposta a un cluster

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

Configurazione della richiesta/risposta da un cluster

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Configurazione dei percorsi dei messaggi tra cluster

Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Ad esempio:

- Un'applicazione al gestore code VENICE invia un messaggio al gestore code PISA utilizzando la chiamata MQPUT . L'applicazione fornisce le seguenti informazioni sulla coda di risposta nel descrittore del messaggio:

```
ReplyToQ='QUEUE'  
ReplyToQMgI=''
```

- Affinché le risposte inviate a QUEUE possano essere ricevute su OTHERQ alle PISA, creare una definizione di coda remota su VENICE utilizzata come alias della coda di risposta. L'alias è valido solo sul sistema su cui è stato creato.

```
DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
```

RQMNAME e QREMOTE possono specificare gli stessi nomi, anche se RQMNAME è un gestore code del cluster.

Concetti correlati

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

Attività correlate

Configurazione della richiesta/risposta a un cluster

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

Configurazione della richiesta/risposta da un cluster

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Configurazione dei percorsi dei messaggi tra cluster

Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

Alias coda e cluster

Utilizzare gli alias della coda per nascondere il nome di una coda cluster, per raggruppare una coda, per adottare attributi differenti o per adottare controlli accessi differenti.

Una definizione QALIAS viene utilizzata per creare un alias mediante il quale una coda deve essere riconosciuta. È possibile creare un alias per una serie di ragioni:

- Si desidera iniziare a utilizzare una coda diversa ma non si desidera modificare le applicazioni.
- Non si desidera che le applicazioni conoscano il nome reale della coda in cui stanno inserendo i messaggi.
- È possibile che si disponga di una convenzione di denominazione diversa da quella in cui è definita la coda.
- Le applicazioni potrebbero non essere autorizzate ad accedere alla coda in base al suo nome reale, ma solo in base all'alias.

Creare una definizione QALIAS su un gestore code utilizzando il comando `DEFINE QALIAS`. Ad esempio, eseguire il comando:

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

Il comando annuncia una coda denominata PUBLIC ai gestori code nel cluster C. PUBLIC è un alias che si risolve nella coda denominata LOCAL. I messaggi inviati a PUBLIC vengono instradati alla coda denominata LOCAL.

È anche possibile utilizzare una definizione alias della coda per risolvere un nome coda in una coda cluster. Ad esempio, eseguire il comando:

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

Il comando consente a un gestore code di utilizzare il nome PRIVATE per accedere a una coda pubblicizzata altrove nel cluster dal nome PUBLIC. Poiché questa definizione non include l'attributo CLUSTER, viene applicata solo al gestore code che la crea.

Concetti correlati

Cluster e alias del gestore code

Utilizzare gli alias del gestore code per nascondere il nome dei gestori code quando si inviano messaggi all'interno o all'esterno di un cluster e per bilanciare il carico di lavoro dei messaggi inviati a un cluster.

Cluster e alias della coda di risposta

Una definizione alias coda di risposta viene utilizzata per specificare nomi alternativi per le informazioni di risposta. Le definizioni di alias della coda di risposta possono essere utilizzate con i cluster esattamente come in un ambiente di accodamento distribuito.

Attività correlate

Configurazione della richiesta/risposta a un cluster

Configurare un percorso del messaggio di richiesta / risposta da un gestore code esterno a un cluster. Nascondere i dettagli interni del cluster utilizzando un gestore code del gateway come percorso di comunicazione verso e dal cluster.

Configurazione della richiesta/risposta da un cluster

Configurare un percorso del messaggio di richiesta / risposta da un cluster a un gestore code esterno al cluster. Nascondere i dettagli su come un gestore code all'interno del cluster comunica all'esterno del cluster utilizzando un gestore code gateway.

Configurazione del bilanciamento del workload dall'esterno di un cluster

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Configurazione dei percorsi dei messaggi tra cluster


Connettere i cluster utilizzando un gestore code gateway. Rendere le code o i gestori code visibili a tutti i cluster definendo gli alias della coda del cluster o del gestore code del cluster sul gestore code del gateway.

Utilizzo dei cluster per la gestione del carico di lavoro

Definendo più istanze di una coda su gestori code differenti in un cluster, è possibile distribuire il lavoro di gestione della coda su più server. Esistono diversi fattori che possono impedire la riaccodamento dei messaggi a un gestore code differente in caso di errore.

Oltre a configurare i cluster per ridurre l'amministrazione del sistema, è possibile creare cluster in cui più di un gestore code ospita un'istanza della stessa coda.

È possibile organizzare il cluster in modo che i gestori code al suo interno siano cloni l'uno dell'altro. Ogni gestore code è in grado di eseguire le stesse applicazioni e dispone di definizioni locali delle stesse code.

 Ad esempio, in un sysplex parallelo z/OS le applicazioni clonate potrebbero accedere ai dati in un database VSAM (Virtual Storage Access Method) o Db2 condiviso. È possibile distribuire il workload tra i propri gestori code disponendo di diverse istanze di un'applicazione. Ogni istanza dell'applicazione riceve messaggi e viene eseguita indipendentemente dalle altre.

I vantaggi di utilizzare i cluster in questo modo sono i seguenti:

- Maggiore disponibilità delle code e delle applicazioni.
- Maggiore velocità di trasmissione dei messaggi.
- Distribuzione più uniforme del carico di lavoro nella rete.

Uno qualsiasi dei gestori code che ospita un'istanza di una particolare coda può gestire i messaggi destinati a tale coda e le applicazioni non denominano un gestore code quando inviano i messaggi. Se un cluster contiene più di un'istanza della stessa coda, IBM MQ seleziona un gestore code a cui instradare un messaggio. Le destinazioni adatte vengono scelte in base alla disponibilità del gestore code e della coda

e in base a un certo numero di attributi specifici del carico di lavoro del cluster associati a gestori code, code e canali. Vedi [Bilanciamento del carico di lavoro in cluster](#).

z/OS In IBM MQ for z/OS, i gestori code che si trovano in gruppi di condivisione code possono ospitare code cluster come code condivise. Le code cluster condivise sono disponibili per tutti i gestori code nello stesso gruppo di condivisione code. Ad esempio, in [Un cluster con più istanze della stessa coda](#), uno o entrambi i gestori code QM2 e QM4 possono essere un gestore code condiviso. Ognuno ha una definizione per la coda Q3. Qualsiasi gestore code nello stesso gruppo di condivisione code di QM4 può leggere un messaggio inserito nella coda condivisa Q3. Ogni gruppo di condivisione code può contenere fino a 32 gestori code, ognuno con accesso agli stessi dati. La condivisione della coda aumenta in modo significativo la velocità di trasmissione dei messaggi.

Consultare i seguenti argomenti secondari per ulteriori informazioni sulle configurazioni cluster per la gestione del carico di lavoro:

Concetti correlati

[Confronto tra cluster e accodamento distribuito](#)

[Accodamento distribuito e cluster](#)

[Componenti di un cluster](#)

[Canali cluster](#)

[Cosa succede se una coda cluster è disabilitata per MQPUT](#)

[Il bilanciamento del carico di lavoro impostato su un canale mittente del cluster non sta funzionando](#)

[“Instradamento dei messaggi verso e dai cluster” a pagina 386](#)

Utilizzare gli alias di coda, gli alias di gestore code e le definizioni di code remote per connettere i cluster a gestori code esterni e altri cluster.

Attività correlate

[Scrittura e compilazione delle uscite del carico di lavoro del cluster](#)

[“Configurazione di un cluster di gestore code” a pagina 303](#)

I cluster forniscono un meccanismo per l'interconnessione dei gestori code in modo da semplificare sia la configurazione iniziale che la gestione in corso. È possibile definire componenti cluster e creare e gestire cluster.

[“Configurazione di un nuovo cluster” a pagina 318](#)

Seguire queste istruzioni per configurare il cluster di esempio. Istruzioni separate descrivono l'impostazione del cluster su TCP/IP, LU 6.2e con una o più code di trasmissione. Verificare il funzionamento del cluster inviando un messaggio da un gestore code all'altro.

[“Configurazione del bilanciamento del workload dall'esterno di un cluster” a pagina 395](#)

Configurare un percorso del messaggio da un gestore code esterno a un cluster a qualsiasi copia di una coda cluster. Il risultato è di bilanciare il carico di lavoro delle richieste dall'esterno del cluster a ciascuna istanza di una coda cluster.

Riferimenti correlati

[Programma di esempio Monitoraggio coda cluster \(AMQSCLM\)](#)

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

[Figura 59 a pagina 407](#) mostra un cluster in cui è presente più di una definizione per la coda Q3.

Se un'applicazione in QM1 inserisce un messaggio in Q3, non necessariamente sa quale istanza di Q3 elaborerà il suo messaggio. Se un'applicazione è in esecuzione su QM2 o QM4, in cui sono presenti istanze locali di Q3, l'istanza locale di Q3 viene aperta per impostazione predefinita. Impostando l'attributo della coda CLWLUSEQ, l'istanza locale della coda può essere trattata come un'istanza remota della coda.

L'opzione MQOPEN DefBind controlla se il gestore code di destinazione viene scelto quando viene emessa la chiamata MQOPEN o quando il messaggio viene trasferito dalla coda di trasmissione.

Se si imposta DefBind su MQBND_BIND_NOT_FIXED , il messaggio può essere inviato a un'istanza della coda disponibile quando il messaggio viene trasmesso. Ciò evita i seguenti problemi:

- La coda di destinazione non è disponibile quando il messaggio arriva al gestore code di destinazione.
- Lo stato della coda è cambiato.
- Il messaggio è stato inserito utilizzando un alias della coda cluster e non esiste alcuna istanza della coda di destinazione sul gestore code in cui è definita l'istanza dell'alias della coda cluster.

Se questi problemi vengono rilevati al momento della trasmissione, viene ricercata un'altra istanza disponibile della coda di destinazione e il messaggio viene reinstradato. Se non sono disponibili istanze della coda, il messaggio viene inserito nella coda di messaggi non recapitabili.

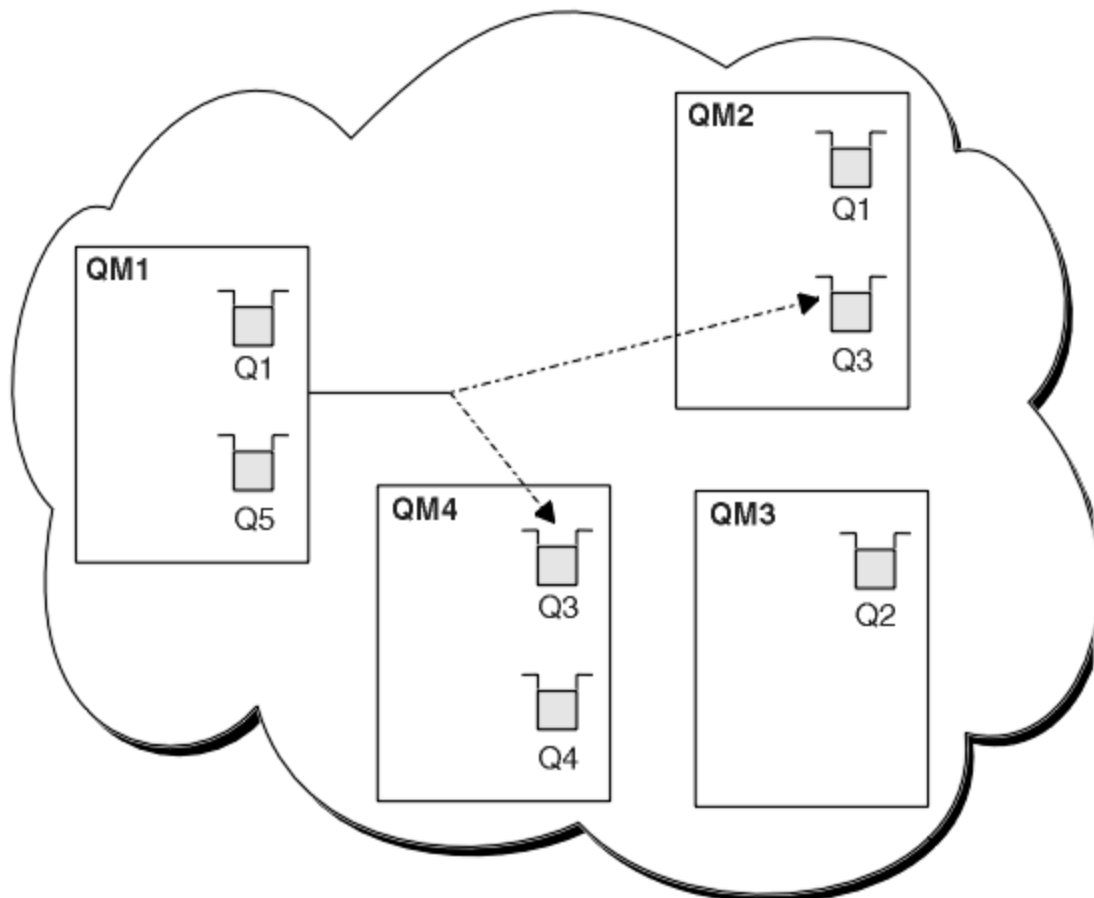



Figura 59. Un cluster con più istanze della stessa coda

Un fattore che può impedire il reinstradamento dei messaggi è se i messaggi sono stati assegnati a un canale o a un gestore code fisso con MQBND_BIND_ON_OPEN. I messaggi collegati su MQOPEN non vengono mai riassegnati ad un altro canale. Si noti inoltre che la riallocazione del messaggio avviene solo quando un canale cluster è in errore. La riassegnazione non si verifica se il canale ha già avuto esito negativo.

Il sistema tenta di reinstradare un messaggio se il gestore code di destinazione non è più in servizio. In questo modo, non influisce sull'integrità del messaggio, correndo il rischio di perderlo o creando un duplicato. Se un gestore code ha esito negativo e lascia un messaggio in dubbio, tale messaggio non viene reinstradato.

z/OS Su IBM MQ for z/OS, il canale non si arresta completamente fino a quando non viene completato il processo di riassegnazione del messaggio. L'interruzione del canale con modalità impostata su FORCE o TERMINATE non interrompe il processo, quindi se si esegue questa operazione, alcuni messaggi BIND_NOT_FIXED potrebbero essere già stati riassegnati a un altro canale oppure i messaggi potrebbero non essere in ordine.

Nota: 

1. Prima di configurare un cluster che abbia più istanze della stessa coda, assicurarsi che i messaggi non abbiano dipendenze tra loro. Ad esempio, deve essere elaborato in una sequenza specifica o dallo stesso gestore code.
2. Rendere identiche le definizioni per istanze differenti della stessa coda. Altrimenti, si ottengono risultati diversi da chiamate MQINQ differenti.

Concetti correlati

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in Aggiunta di un nuovo gestore code a un cluster. Contiene tre gestori code; LONDON e NEWYORK contengono entrambi repository completi, PARIS contiene un repository parziale. L'applicazione di inventario viene eseguita sul sistema a New York, connesso al gestore code NEWYORK. L'applicazione è guidata dall'arrivo di messaggi sulla coda INVENTQ.
- Si desidera aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Informazioni su questa attività

Seguire questa procedura per aggiungere un gestore code che ospita localmente una coda.

Procedura

1. Modificare il gestore code PARIS .

Affinché l'applicazione a Parigi utilizzi INVENTQ a Parigi e quella a New York, è necessario informare il gestore code. Su PARIS immettere il seguente comando:

```
ALTER QMGR CLWLUSEQ(ANY)
```

2. Esaminare l'applicazione inventario per le affinità dei messaggi.


Prima di procedere, assicurarsi che l'applicazione di inventario non abbia alcuna dipendenza dalla sequenza di elaborazione dei messaggi. Per ulteriori informazioni, consultare [Gestione delle affinità dei messaggi](#).

3. Installare l'applicazione inventario sul sistema a Parigi.

4. Definire la coda del cluster INVENTQ.

La coda INVENTQ già ospitata dal gestore code NEWYORK deve essere ospitata anche da PARIS. Definirlo sul gestore code PARIS nel modo seguente:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

 Ora che sono state completate tutte le definizioni, se non è già stato fatto, avviare l'iniziatore di canali su IBM MQ for z/OS.

Su tutte le piattaforme, avviare un programma listener sul Gestore code PARIS. Il listener ascolta le richieste di rete in entrata e avvia il canale ricevente del cluster quando è necessario.

Risultati

[Figura 60 a pagina 409](#) mostra il cluster impostato da questa attività.

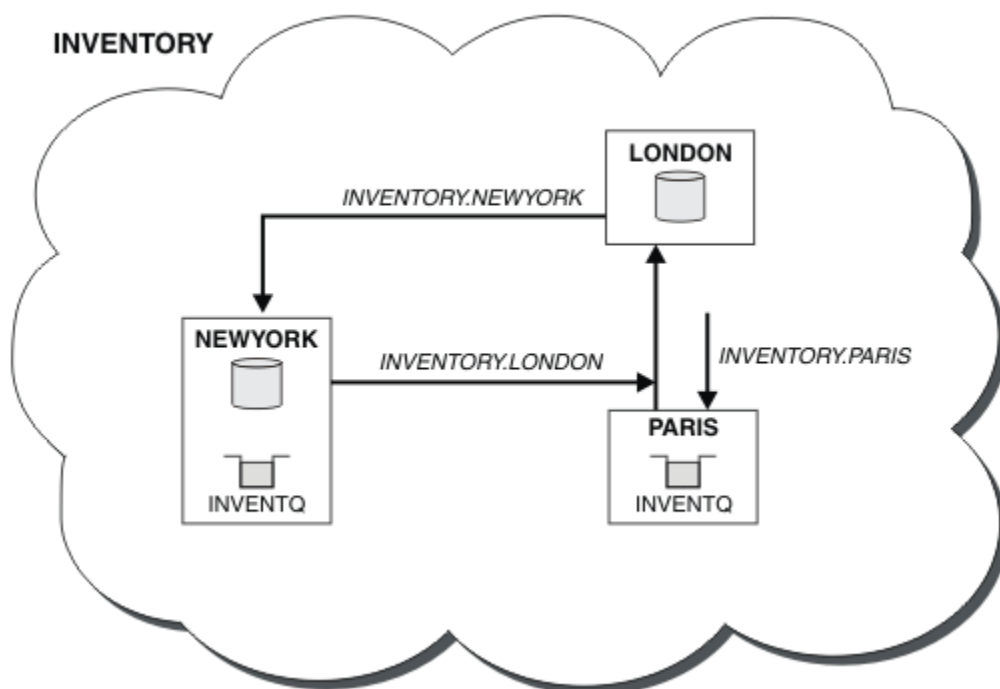


Figura 60. Il cluster INVENTORY, con tre gestori code

La modifica a questo cluster è stata effettuata senza modificare i gestori code NEWYORK o LONDON. I repository completi in questi gestori code vengono aggiornati automaticamente con le informazioni necessarie per inviare messaggi a INVENTQ all'indirizzo PARIS.

Operazioni successive

La coda INVENTQ e l'applicazione di inventario si trovano ora su due gestori code nel cluster. Ciò aumenta la loro disponibilità, velocizza la velocità di trasmissione dei messaggi e consente la distribuzione del carico di lavoro tra i due gestori code. I messaggi immessi in INVENTQ da uno qualsiasi dei gestori code LONDON, NEWYORK, PARIS vengono instradati alternativamente a PARIS o NEWYORK, in modo che il carico di lavoro sia bilanciato.

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in "Aggiunta di un gestore code a un cluster". Contiene tre gestori code; LONDON e NEWYORK contengono entrambi repository completi, PARIS contiene un repository parziale. L'applicazione di inventario viene eseguita sul sistema a New York, connesso al gestore code NEWYORK. L'applicazione è guidata dall'arrivo di messaggi sulla coda INVENTQ.

- È stato aggiunto un nuovo negozio in TOKYO dove sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Informazioni su questa attività

Seguire questa procedura per utilizzare due reti in un cluster.

Procedura

1. Decidere quale repository completo TOKYO fa riferimento per primo.

Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi per raccogliere informazioni sul cluster. Crea il proprio repository parziale. Non è di particolare importanza quale repository si sceglie. In questo esempio, viene selezionato NEWYORK . Una volta che il nuovo gestore code si è unito al cluster, comunica con entrambi i repository.

2. Definire i canali CLUSRCVR .

Ogni gestore code in un cluster deve definire un ricevente del cluster su cui può ricevere messaggi. Questo gestore code deve essere in grado di comunicare su ciascuna rete.


```
DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network B for TOKYO')
```

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network A for TOKYO')
```

3. Definire un canale CLUSSDR nel gestore code TOKYO .

Ogni gestore code di un cluster deve definire un canale mittente del cluster su cui può inviare messaggi al primo repository completo. In questo caso è stato scelto NEWYORK, quindi TOKYO ha bisogno della definizione seguente:

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from TOKYO to repository at NEWYORK')
```

 Ora che sono state completate tutte le definizioni, se non è stato ancora fatto, avviare l'iniziatore di canali su IBM MQ for z/OS.

Su tutte le piattaforme, avviare un programma listener sul Gestore code PARIS. Il programma listener ascolta le richieste di rete in entrata e avvia il canale ricevente del cluster quando è necessario.

Risultati

[Figura 61 a pagina 412](#) mostra il cluster impostato da questa attività.

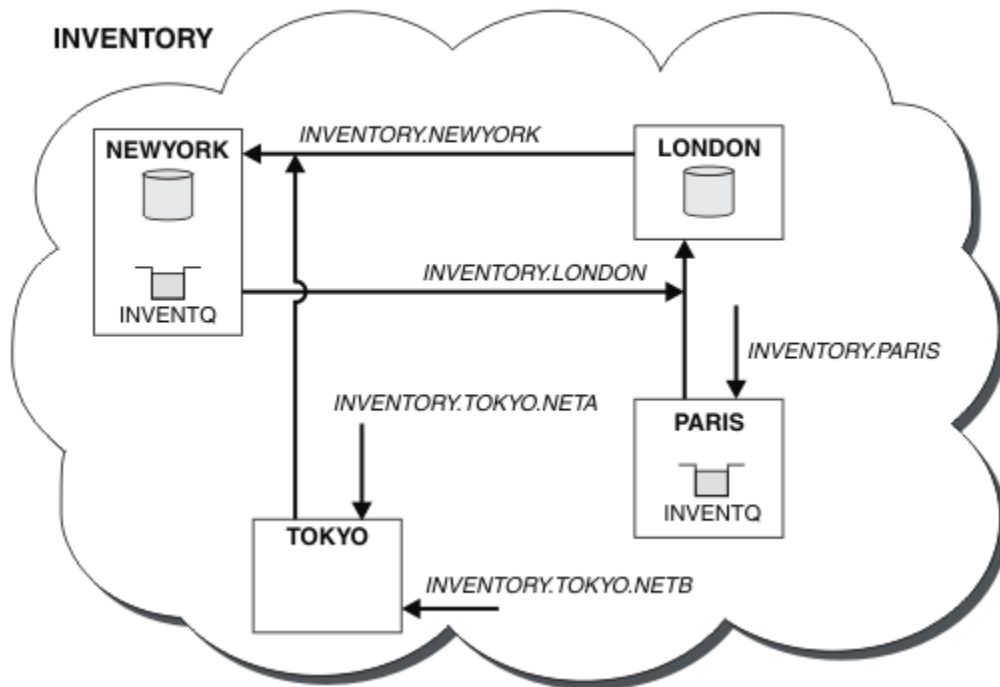


Figura 61. Il cluster INVENTORY , con quattro gestori code

Facendo solo tre definizioni, abbiamo aggiunto il gestore code TOKYO al cluster con due diversi instradamenti di rete disponibili.

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

“Aggiunta di un gestore code a un cluster” a pagina 329

Seguire queste istruzioni per aggiungere il gestore code al cluster creato. I messaggi per le code e gli argomenti del cluster vengono trasferiti utilizzando la singola coda di trasmissione del cluster SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in “Utilizzo di due reti in un cluster” a pagina 410. Contiene quattro gestori code; LONDON e NEWYORK contengono entrambi repository completi; PARIS e TOKYO conservano repository parziali. L'applicazione di inventario viene eseguita sul sistema a New York, connesso al gestore code NEWYORK. Il gestore code TOKYO ha due reti differenti su cui può comunicare.
- Si desidera rendere una delle reti la rete primaria e un'altra delle reti la rete di backup. Si prevede di utilizzare la rete di backup se si verifica un problema con la rete principale.

Informazioni su questa attività

Utilizzare l'attributo NETPRTY per configurare una rete primaria e una secondaria in un cluster.

Procedura

Modificare i canali CLUSRCVR esistenti su TOKYO.

Per indicare che il canale di rete A è il canale primario e il canale di rete B è il canale secondario, utilizzare i seguenti comandi:

- a) ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')
- b) ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1) DESCR('Backup cluster-receiver channel for TOKYO')

Operazioni successive

Configurando il canale con priorità di rete differenti, hai ora definito per il cluster che hai una rete primaria e una rete secondaria. I gestori code nel cluster che utilizzano questi canali utilizzano automaticamente la rete primaria quando è disponibile. I gestori code eseguono il failover per utilizzare la rete secondaria quando la rete primaria non è disponibile.

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in “Aggiunta di un gestore code a un cluster” a pagina 329. Contiene tre gestori code; LONDON e NEWYORK contengono entrambi repository completi, PARIS contiene un repository parziale. L'applicazione di inventario viene eseguita sul sistema a New York, connesso al gestore code NEWYORK . L'applicazione è guidata dall'arrivo di messaggi sulla coda INVENTQ .
- Un nuovo negozio è stato creato a Chicago per fornire un backup per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Informazioni su questa attività

Effettuare le operazioni riportate di seguito per aggiungere una coda da utilizzare come backup.

Procedura

1. Decidere quale repository completo CHICAGO fa riferimento per primo.

Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi per raccogliere informazioni sul cluster. Crea il proprio repository parziale. Non è di particolare importanza quale repository si sceglie per un particolare gestore code. In questo esempio, viene selezionato NEWYORK . Una volta che il nuovo gestore code si è unito al cluster, comunica con entrambi i repository.

2. Definire il canale CLUSRCVR .

Ogni gestore code in un cluster deve definire un ricevente del cluster su cui può ricevere messaggi. Su CHICAGO, definire:

```
DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel for CHICAGO')
```

3. Definire un canale CLUSSDR nel gestore code CHICAGO.

Ogni gestore code di un cluster deve definire un canale mittente del cluster su cui può inviare messaggi al primo repository completo. In questo caso è stato scelto NEWYORK, quindi CHICAGO ha bisogno della definizione seguente:

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from CHICAGO to repository at NEWYORK')
```

4. Modificare la coda cluster esistente INVENTQ.

INVENTQ che è già ospitato dal gestore code NEWYORK è l'istanza principale della coda.

```
ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
```

5. Esaminare l'applicazione inventario per le affinità dei messaggi.


Prima di procedere, assicurarsi che l'applicazione di inventario non abbia alcuna dipendenza dalla sequenza di elaborazione dei messaggi.

6. Installare l'applicazione di inventario sul sistema in CHICAGO.

7. Definire la coda del cluster di backup INVENTQ

Il INVENTQ che è già ospitato dal gestore code NEWYORK , deve essere ospitato anche come backup da CHICAGO. Definirlo sul gestore code CHICAGO nel modo seguente:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
```

 Ora che sono state completate tutte le definizioni, se non è stato ancora fatto, avviare l'inziatore di canali su IBM MQ for z/OS.

Su tutte le piattaforme, avviare un programma listener sul Gestore code CHICAGO. Il programma listener ascolta le richieste di rete in entrata e avvia il canale ricevente del cluster quando è necessario.

Risultati

[Figura 62 a pagina 416](#) mostra il cluster impostato da questa attività.

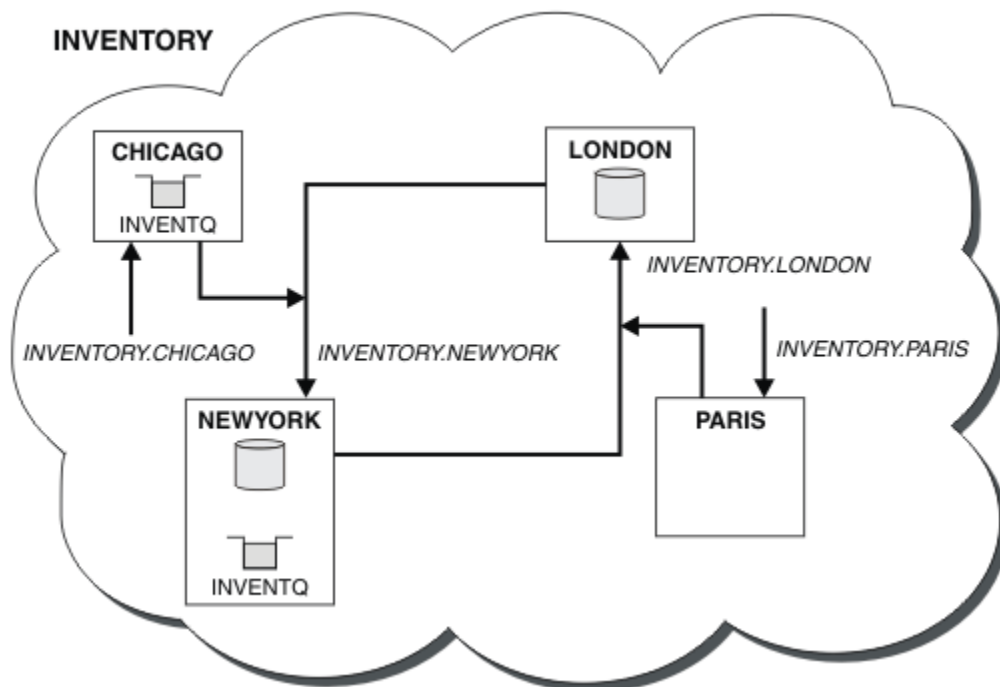


Figura 62. Il cluster INVENTORY, con quattro gestori code

La coda INVENTQ e l'applicazione di inventario si trovano ora su due gestori code nel cluster. Il gestore code CHICAGO è un backup. I messaggi immessi in INVENTQ vengono instradati a NEWYORK a meno che non siano non disponibili quando vengono inviati a CHICAGO.

Nota:

La disponibilità di un gestore code remoto si basa sullo stato del canale per tale gestore code. Quando i canali vengono avviati, il loro stato cambia diverse volte, con alcuni degli stati meno preferibili all'algoritmo di gestione del carico di lavoro del cluster. In pratica, ciò significa che è possibile scegliere destinazioni con priorità più bassa (backup) mentre i canali verso destinazioni con priorità più alta (primaria) sono in fase di avvio.

Se è necessario accertarsi che nessun messaggio venga inviato a una destinazione di backup, non utilizzare CLWLPRTY. Prendere in considerazione l'utilizzo di code separate o di CLWLRANK con un passaggio manuale dal primario al backup.

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Un'applicazione di controllo prezzi deve essere installata su vari gestori code. Per mantenere basso il numero di canali utilizzati, il numero di canali attivi eseguiti da ciascun server è limitato. L'applicazione è guidata dall'arrivo di messaggi sulla coda PRICEQ .
- Quattro gestori code del server ospitano l'applicazione di controllo prezzi. Due gestori code di query inviano messaggi a PRICEQ per interrogare un prezzo. Altri due gestori code sono configurati come repository completi.

Informazioni su questa attività

Effettuare le operazioni riportate di seguito per limitare il numero di canali utilizzati.

Procedura

1. Scegliere due repository completi.

Scegliere due gestori code come repository completi per il cluster di controllo prezzi. Sono denominati REPOS1 e REPOS2.

Emetti il seguente comando:

```
ALTER QMGR REPOS(PRICECHECK)
```

2. Definire un canale CLUSRCVR su ciascun gestore code.

In ogni gestore code del cluster, definire un canale ricevente del cluster e un canale mittente del cluster. Non importa quale sia definito per primo.

```
DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
```

3. Definire un canale CLUSSDR su ciascun gestore code.

Creare una definizione CLUSSDR su ogni gestore code per collegare tale gestore code a uno o più gestori code del repository completo.

```
DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to  
repository queue manager')
```

4. Installare l'applicazione di controllo prezzi.
5. Definire la coda PRICEQ su tutti i gestori code server.

Immettere il seguente comando per ciascuno di essi:

```
DEFINE QLOCAL (PRICEQ) CLUSTER (PRICECHECK)
```

6. Limitare il numero di canali utilizzati dalle query

Sui gestori code delle query viene limitato il numero di canali attivi utilizzati, immettendo i seguenti comandi su ciascuno di essi:

```
ALTER QMGR CLWLMRUC (2)
```

7. Avviare un programma listener.

z/OS Se non è stato ancora fatto, avviare l'iniziatore di canali su IBM MQ for z/OS.

Su tutte le piattaforme, avviare un programma listener. Il programma listener ascolta le richieste di rete in entrata e avvia il canale ricevente del cluster quando è necessario.

Risultati

Figura 63 a pagina 418 mostra il cluster impostato da questa attività.

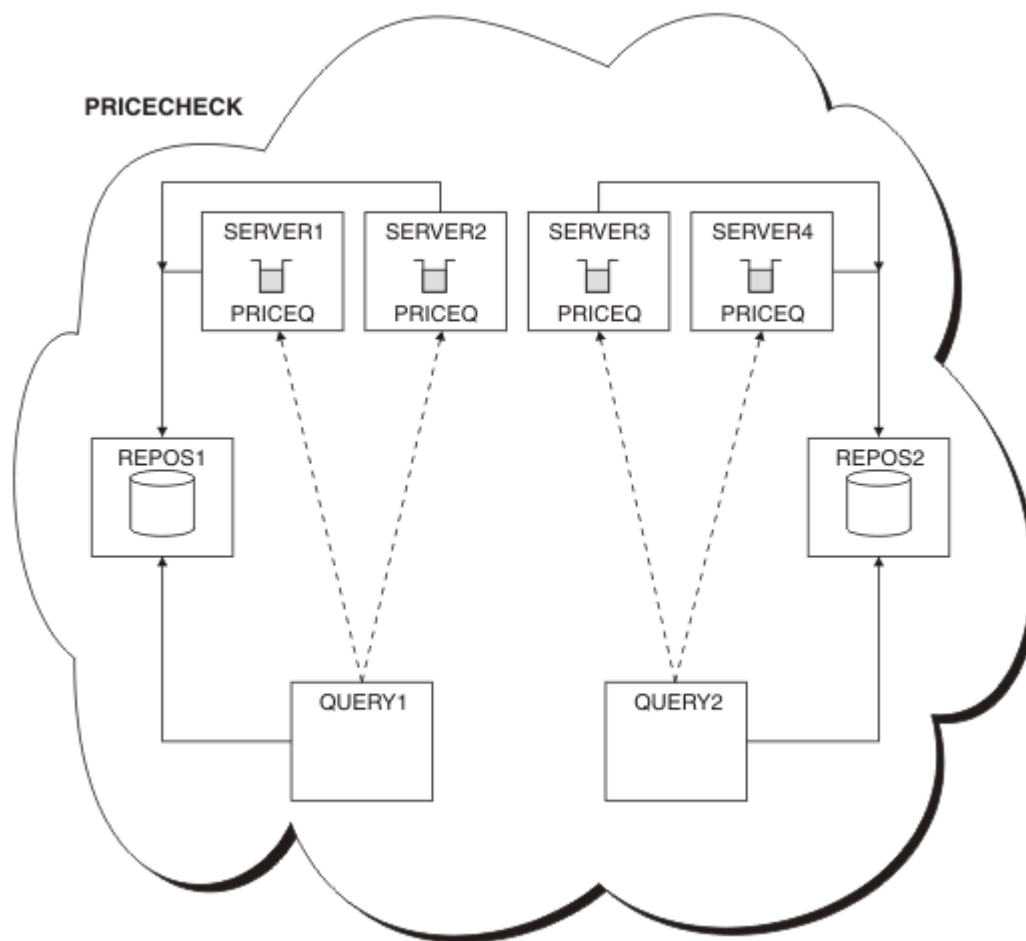


Figura 63. Il cluster PRICECHECK , con quattro gestori code del server, due repository e due gestori code di query

Anche se ci sono quattro istanze della coda PRICEQ disponibili nel cluster PRICECHECK , ogni gestore code che esegue la query utilizza solo due di esse. Ad esempio, il gestore code QUERY1 dispone solo di canali attivi per i gestori code SERVER1 e SERVER2 . Se SERVER1 diventasse non disponibile, il gestore code QUERY1 inizierebbe a utilizzare un altro gestore code, ad esempio SERVER3.

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Prima di iniziare

Nota: Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in “Aggiunta di un gestore code a un cluster” a pagina 329. Contiene tre gestori code: LONDON e NEWYORK contengono entrambi repository completi, PARIS contiene un repository parziale e inserisce i messaggi da INVENTQ. L'applicazione di inventario viene eseguita sul sistema a New York connesso al gestore code NEWYORK. L'applicazione è guidata dall'arrivo di messaggi sulla coda INVENTQ.
- Si sta allestendo un nuovo negozio a Los Angeles. Per fornire capacità aggiuntiva, si desidera eseguire il sistema di inventario a Los Angeles e New York. Il nuovo gestore code può elaborare il doppio dei messaggi rispetto a New York.

Informazioni su questa attività

Seguire questa procedura per aggiungere un gestore code più potente che ospita una coda.

Procedura

1. Decidere quale repository completo LOSANGELES fa riferimento per primo.

2. Ogni gestore code in un cluster deve fare riferimento a uno o più repository completi per raccogliere informazioni sul cluster. Crea il proprio repository parziale. Non è di particolare importanza quale repository si sceglie. In questo esempio, viene selezionato NEWYORK . Una volta che il nuovo gestore code si è unito al cluster, comunica con entrambi i repository.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
```

3. Definire il canale CLUSRCVR sul gestore code LOSANGELES.

Ogni gestore code in un cluster deve definire un canale ricevente del cluster su cui può ricevere i messaggi. Su LOSANGELES, definire:

```
DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager LOSANGELES')
CLWLWGHT(2)
```

Il canale ricevente del cluster annuncia la disponibilità del gestore code a ricevere messaggi da altri gestori code nel cluster INVENTORY. L'impostazione di CLWLWGHT su due garantisce che il gestore code di Los Angeles riceva il doppio dei messaggi di inventario rispetto a New York (quando il canale di NEWYORK è impostato su uno).

4. Modificare il canale CLUSRCVR sul gestore code NEWYORK.

Assicurarsi che il gestore code di Los Angeles riceva il doppio dei messaggi di inventario di New York. Modificare la definizione del canale ricevente del cluster.

```
ALTER CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) CLWLWGHT(1)
```

5. Esaminare l'applicazione inventario per le affinità dei messaggi.


Prima di procedere, assicurarsi che l'applicazione di inventario non abbia alcuna dipendenza dalla sequenza di elaborazione dei messaggi.

6. Installa l'applicazione di inventario sul sistema a Los Angeles

7. Definire la coda del cluster INVENTQ.

La coda INVENTQ , che è già ospitata dal gestore code NEWYORK , deve essere ospitata anche da LOSANGELES. Definirlo sul gestore code LOSANGELES nel modo seguente:

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

 Ora che sono state completate tutte le definizioni, se non è stato ancora fatto, avviare l'iniziatore di canali su IBM MQ for z/OS.

Su tutte le piattaforme, avviare un programma listener sul Gestore code LOSANGELES. Il programma listener ascolta le richieste di rete in entrata e avvia il canale ricevente del cluster quando è necessario.

Risultati

“[Aggiunta di un gestore code più potente che ospita una coda](#)” a pagina 419 mostra il cluster impostato da questa attività.

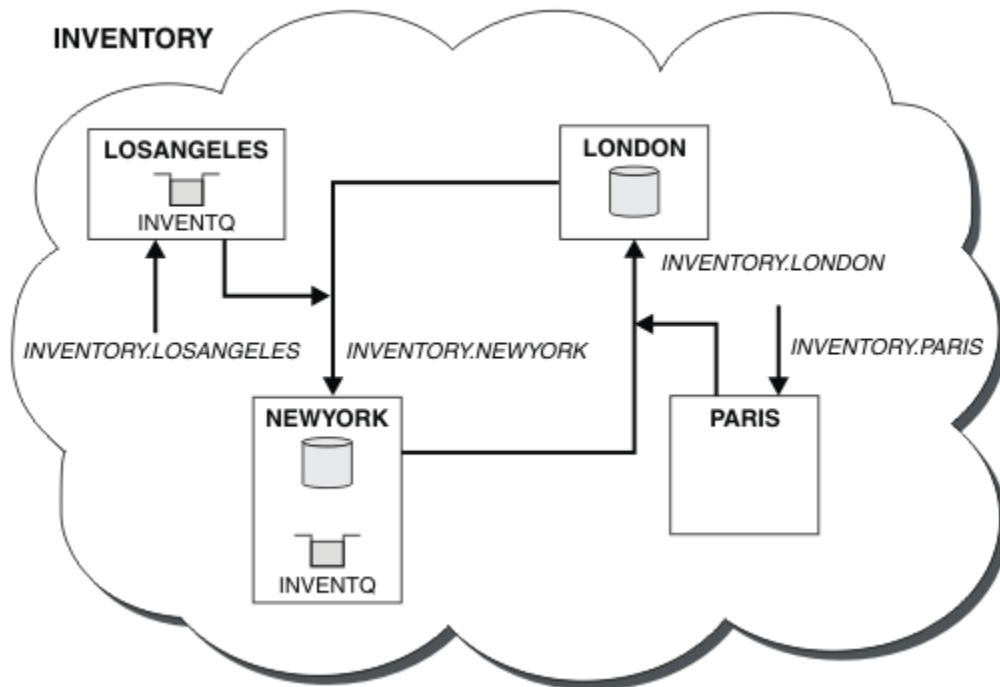


Figura 64. Il cluster INVENTORY con quattro gestori code

Questa modifica al cluster è stata effettuata senza dover modificare i gestori code LONDON e PARIS. I repository in questi gestori code vengono aggiornati automaticamente con le informazioni necessarie per inviare messaggi a INVENTQ all'indirizzo LOSANGELES.

Operazioni successive

La coda INVENTQ e l'applicazione di inventario si trovano su due gestori code nel cluster. La configurazione aumenta la loro disponibilità, velocizza la velocità di trasmissione dei messaggi e consente la distribuzione del carico di lavoro tra i due gestori code. I messaggi inseriti in INVENTQ da LOSANGELES o NEWYORK vengono gestiti dall'istanza sul gestore code locale quando possibile. I messaggi inseriti da LONDON o PARIS vengono instradati a LOSANGELES o NEWYORK, con il doppio dei messaggi inviati a LOSANGELES.

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Cluster e programmazione delle applicazioni

Non è necessario apportare alcuna modifica di programmazione per trarre vantaggio da più istanze della stessa coda. Tuttavia, alcuni programmi non funzionano correttamente a meno che non venga inviata una sequenza di messaggi alla stessa istanza di una coda.

Le applicazioni possono aprire una coda utilizzando la chiamata MQOPEN . Le applicazioni utilizzano la chiamata MQPUT per inserire messaggi in una coda aperta. Le applicazioni possono inserire un singolo messaggio in una coda non già aperta, utilizzando la chiamata MQPUT1 .

Se si configurano i cluster che hanno più istanze della stessa coda, non ci sono considerazioni specifiche sulla programmazione dell'applicazione. Tuttavia, per trarre vantaggio dagli aspetti di gestione del carico di lavoro del cluster, potrebbe essere necessario modificare le applicazioni. Se si imposta una rete in cui sono presenti più definizioni della stessa coda, esaminare le applicazioni per le affinità dei messaggi.

Si supponga, ad esempio, di avere due applicazioni che si basano su una serie di messaggi che scorrono tra di loro sotto forma di domande e risposte. Probabilmente si desidera che le risposte ritornino allo stesso gestore code che ha inviato una domanda. È importante che la routine di gestione del carico di lavoro non invii i messaggi ad alcun gestore code che ospita una copia della coda di risposta.

È possibile disporre di applicazioni che richiedono l'elaborazione dei messaggi in sequenza (ad esempio, un'applicazione di replica del database che invia batch di messaggi che devono essere richiamati in sequenza). L'utilizzo di messaggi segmentati può anche causare un problema di affinità.

Apertura di una versione locale o remota della coda di destinazione

Tenere presente il modo in cui il gestore code sceglie di utilizzare una versione locale o remota della coda di destinazione.

1. Il gestore code apre la versione locale della coda di destinazione per leggere i messaggi o per impostare gli attributi della coda.
2. Il gestore code apre qualsiasi istanza della coda di destinazione in cui scrivere i messaggi, se si verifica almeno una delle seguenti condizioni:
 - Una versione locale della coda di destinazione non esiste.
 - Il gestore code specifica CLWLUSEQ (ANY) su ALTER QMGR.
 - La coda sul gestore code specifica CLWLUSEQ (ANY) .

Concetti correlati

Esempio di un cluster con più di un'istanza di una coda

In questo esempio di cluster con più di un'istanza di una coda, i messaggi vengono instradati a istanze differenti della coda. È possibile forzare un messaggio a una specifica istanza della coda ed è possibile scegliere di inviare una sequenza di messaggi a uno dei gestori code.

Attività correlate

Aggiunta di un gestore code che ospita una coda localmente

Seguire queste istruzioni per aggiungere un'istanza di INVENTQ per fornire ulteriore capacità per eseguire il sistema dell'applicazione di inventario a Parigi e New York.

Utilizzo di due reti in un cluster

Seguire queste istruzioni per aggiungere un nuovo negozio in TOKYO in cui sono presenti due reti differenti. Entrambi devono essere disponibili per comunicare con il gestore code di Tokyo.

Utilizzo di una rete primaria e di una secondaria in un cluster

Seguire queste istruzioni per rendere una rete la rete principale e un'altra la rete di backup. Utilizzare la rete di backup se si verifica un problema con la rete principale.

Aggiunta di una coda da utilizzare come backup

Seguire queste istruzioni per fornire un backup a Chicago per il sistema di inventario che ora viene eseguito a New York. Il sistema di Chicago è usato solo quando c'è un problema con il sistema di New York.

Limitazione del numero di canali utilizzati

Seguire queste istruzioni per limitare il numero di canali attivi che ciascun server esegue quando un'applicazione di controllo prezzi è installata su vari gestori code.

Aggiunta di un gestore code più potente che ospita una coda

Seguire queste istruzioni per fornire ulteriore capacità eseguendo il sistema di inventario a Los Angeles e New York, dove Los Angeles può gestire il doppio del numero di messaggi rispetto a New York.

Gestione delle affinità dei messaggi

Le affinità dei messaggi sono raramente parte di un buon progetto di programmazione. È necessario rimuovere le affinità di messaggi per utilizzare completamente il clustering. Se non è possibile rimuovere le affinità dei messaggi, è possibile forzare la consegna dei messaggi correlati utilizzando lo stesso canale e lo stesso gestore code.

Se si dispone di applicazioni con affinità di messaggi, rimuovere le affinità prima di iniziare a utilizzare i cluster.

La rimozione delle affinità dei messaggi migliora la disponibilità delle applicazioni. Un'applicazione invia un batch di messaggi con affinità di messaggi a un gestore code. Il gestore code non riesce dopo aver ricevuto solo una parte del batch. Il gestore code di invio deve attendere il ripristino ed elaborare il batch di messaggi incompleto prima di poter inviare ulteriori messaggi.

La rimozione delle affinità dei messaggi migliora anche la scalabilità delle applicazioni. Un batch di messaggi con affinità può bloccare le risorse sul gestore code di destinazione in attesa di messaggi successivi. Queste risorse potrebbero rimanere bloccate per lunghi periodi di tempo, impedendo ad altre applicazioni di svolgere il proprio lavoro.

Inoltre, le affinità dei messaggi impediscono alle routine di gestione del carico di lavoro del cluster di effettuare la scelta migliore del gestore code.

Per rimuovere le affinità, considerare le seguenti possibilità:

- Trasmissione delle informazioni di stato nei messaggi
- Gestione delle informazioni sullo stato nella memoria non volatile accessibile a qualsiasi gestore code, ad esempio in un database Db2
- Replica dei dati di sola lettura in modo che siano accessibili a più di un gestore code

Se non è appropriato modificare le proprie applicazioni per rimuovere le affinità dei messaggi, esistono diverse soluzioni possibili per il problema.

Denominare una destinazione specifica sulla chiamata MQOPEN

Specificare il nome della coda remota e il nome del gestore code in ogni chiamata MQOPEN e tutti i messaggi inseriti nella coda utilizzando l'handle dell'oggetto vanno allo stesso gestore code, che potrebbe essere il gestore code locale.

La specifica del nome della coda remota e del gestore code su ogni chiamata MQOPEN presenta degli svantaggi:

- Non viene eseguito alcun bilanciamento del workload. Non si traggono vantaggi dal bilanciamento del carico di lavoro del cluster.

- Se il gestore code di destinazione è remoto e vi è più di un canale, i messaggi potrebbero essere instradati in modo diverso e la sequenza di messaggi non viene ancora conservata.
- Se il gestore code dispone di una definizione per una coda di trasmissione con lo stesso nome del gestore code di destinazione, i messaggi vengono inseriti in tale coda di trasmissione piuttosto che nella coda di trasmissione del cluster.

Restituisce il nome del gestore code nel campo del gestore code di risposta

Consentire al gestore code che riceve il primo messaggio in batch di restituire il suo nome nella risposta. Ciò avviene utilizzando il campo `ReplyToQMGr` del descrittore del messaggio. Il gestore code all'estremità di invio può quindi estrarre il nome del gestore code di risposta e specificarlo su tutti i messaggi successivi.

L'uso delle informazioni `ReplyToGestore code` dalla risposta ha degli svantaggi:

- Il gestore code richiedente deve attendere una risposta al primo messaggio
- È necessario scrivere ulteriore codice per trovare e utilizzare le informazioni `ReplyToGestore code` prima di inviare i messaggi successivi
- Se è presente più di un instradamento al gestore code, la sequenza dei messaggi potrebbe non essere conservata

Impostare l'opzione `MQ00_BIND_ON_OPEN` sulla chiamata `MQOPEN`

Forzare tutti i messaggi da inserire nella stessa destinazione utilizzando l'opzione `MQ00_BIND_ON_OPEN` sulla chiamata `MQOPEN`. È necessario specificare `MQ00_BIND_ON_OPEN` o `MQ00_BIND_ON_GROUP` quando si utilizzano gruppi di messaggi con cluster per garantire che tutti i messaggi nel gruppo vengano elaborati alla stessa destinazione.

Aperto una coda e specificando `MQ00_BIND_ON_OPEN`, si forza l'invio di tutti i messaggi inviati a questa coda alla stessa istanza della coda. `MQ00_BIND_ON_OPEN` esegue il bind di tutti i messaggi allo stesso gestore code e allo stesso instradamento. Ad esempio, se esiste un instradamento IP e un instradamento NetBIOS alla stessa destinazione, uno di questi viene selezionato quando la coda viene aperta e questa selezione viene rispettata per tutti i messaggi inseriti nella stessa coda utilizzando l'handle dell'oggetto ottenuto.

Specificando `MQ00_BIND_ON_OPEN` si forzano tutti i messaggi ad essere instradati alla stessa destinazione. Pertanto, le applicazioni con affinità di messaggi non vengono interrotte. Se la destinazione non è disponibile, i messaggi rimangono nella coda di trasmissione fino a quando non diventano nuovamente disponibili.

`MQ00_BIND_ON_OPEN` si applica anche quando il nome gestore code viene specificato nel descrittore oggetto quando si apre una coda. È possibile che vi sia più di un instradamento al gestore code indicato. Ad esempio, potrebbero essere presenti più percorsi di rete oppure un altro gestore code potrebbe aver definito un alias. Se si specifica `MQ00_BIND_ON_OPEN`, viene selezionato un instradamento quando la coda viene aperta.

Nota: Questa è la tecnica consigliata. Tuttavia, non funziona in una configurazione multi - hop in cui un gestore code annuncia un alias per una coda cluster. Inoltre, non è utile nelle situazioni in cui le applicazioni utilizzano code differenti sullo stesso gestore code per diversi gruppi di messaggi.

Un'alternativa per specificare `MQ00_BIND_ON_OPEN` sulla chiamata `MQOPEN`, è quella di modificare le proprie definizioni di coda. Sulle definizioni della coda, specificare `DEFBIND (OPEN)` e consentire l'opzione `DefBind` sulla chiamata `MQOPEN` per impostazione predefinita `MQ00_BIND_AS_Q_DEF`.

Impostare l'opzione `MQ00_BIND_ON_GROUP` sulla chiamata `MQOPEN`

Forzare l'inserimento di tutti i messaggi in un gruppo nella stessa destinazione utilizzando l'opzione `MQ00_BIND_ON_GROUP` nella chiamata `MQOPEN`. È necessario specificare `MQ00_BIND_ON_OPEN` o `MQ00_BIND_ON_GROUP` quando si utilizzano gruppi di messaggi con cluster per garantire che tutti i messaggi nel gruppo vengano elaborati alla stessa destinazione.

Aperto una coda e specificando MQ00_BIND_ON_GROUP, si forza l'invio di tutti i messaggi in un gruppo inviati a questa coda alla stessa istanza della coda. MQ00_BIND_ON_GROUP associa tutti i messaggi in un gruppo allo stesso gestore code e anche allo stesso instradamento. Ad esempio, se c'è un instradamento IP e un instradamento NetBIOS alla stessa destinazione, uno di questi viene selezionato quando la coda viene aperta e questa selezione viene rispettata per tutti i messaggi in un gruppo inserito nella stessa coda utilizzando l'handle dell'oggetto ottenuto.

Specificando MQ00_BIND_ON_GROUP si forzano tutti i messaggi in un gruppo ad essere instradati alla stessa destinazione. Pertanto, le applicazioni con affinità di messaggi non vengono interrotte. Se la destinazione non è disponibile, i messaggi rimangono nella coda di trasmissione fino a quando non diventano nuovamente disponibili.

MQ00_BIND_ON_GROUP si applica anche quando il nome gestore code viene specificato nel descrittore oggetto quando si apre una coda. È possibile che vi sia più di un instradamento al gestore code indicato. Ad esempio, potrebbero essere presenti più percorsi di rete oppure un altro gestore code potrebbe aver definito un alias. Se si specifica MQ00_BIND_ON_GROUP, viene selezionato un instradamento quando la coda viene aperta.

Perché MQ00_BIND_ON_GROUP sia effettivo, è necessario includere l'opzione di inserimento MQPMO_LOGICAL_ORDER in MQPUT. È possibile impostare **GroupId** in MQMD del messaggio su MQGI_NONE ed è necessario includere i seguenti indicatori di messaggio nel campo MQMD **MsgFlags** dei messaggi:

- Ultimo messaggio nel gruppo: MQMF_LAST_MSG_IN_GROUP
- Tutti gli altri messaggi nel gruppo: MQMF_MSG_IN_GROUP

Se MQ00_BIND_ON_GROUP è specificato ma i messaggi non sono raggruppati, il funzionamento è equivalente a MQ00_BIND_NOT_FIXED.

Nota: Questa è la tecnica consigliata per garantire che i messaggi in un gruppo vengano inviati alla stessa destinazione. Tuttavia, non funziona in una configurazione multi-hop in cui un gestore code annuncia un alias per una coda cluster.

Un'alternativa per specificare MQ00_BIND_ON_GROUP sulla chiamata MQOPEN, è quella di modificare le proprie definizioni di coda. Sulle definizioni della coda, specificare DEFBIND (GROUP) e consentire l'opzione DefBind sulla chiamata MQOPEN per impostazione predefinita MQ00_BIND_AS_Q_DEF.

Scrivere un programma di uscita del carico di lavoro del cluster personalizzato

Invece di modificare le applicazioni, è possibile aggirare il problema di affinità dei messaggi scrivendo un programma di uscita del carico di lavoro cluster. La scrittura di un programma di uscita del carico di lavoro del cluster non è semplice e non è una soluzione consigliata. Il programma dovrebbe essere progettato per riconoscere l'affinità ispezionando il contenuto dei messaggi. Dopo aver riconosciuto l'affinità, il programma dovrà forzare il programma di utilità di gestione del carico di lavoro ad instradare tutti i messaggi correlati allo stesso gestore code.

Multi Configurazione di un cluster uniforme

I cluster uniformi consentono alle applicazioni di essere progettate per la scalabilità e la disponibilità e possono connettersi a qualsiasi gestore code all'interno di tale cluster uniforme.

Prima di iniziare

Per un'introduzione al cluster, consultare Cluster. Per un'introduzione ai cluster uniformi, consultare “Informazioni sui cluster uniformi” a pagina 426.

Informazioni su questa attività

I cluster uniformi utilizzano il clustering IBM MQ per la comunicazione tra gestori code e il bilanciamento del carico di lavoro tra code. Tuttavia, differiscono dai tipici cluster IBM MQ nei modi seguenti:


- I cluster uniformi in genere hanno un minor numero di gestori code nel cluster. Non creare un cluster uniforme con più di 10 gestori code.
- Ogni membro del cluster ha una configurazione quasi identica.
- Il cluster viene generalmente utilizzato da una singola applicazione o da un gruppo di applicazioni correlate.
- Il numero di istanze dell'applicazione che si collegano al cluster deve essere maggiore o uguale al numero di gestori code.

È possibile semplificare la creazione di un cluster uniforme e, successivamente, mantenere la configurazione tra i membri del cluster uniforme identici, utilizzando la configurazione automatica e il supporto del cluster automatico.

Procedura

- [Ulteriori informazioni sui cluster uniformi](#)
- [Creare un cluster uniforme](#)
- [Creare un cluster uniforme](#)
- [Sospendi un gestore code da un cluster uniforme](#)

Multi Informazioni sui cluster uniformi

L'obiettivo di una distribuzione cluster uniforme è che le applicazioni possano essere progettate per la scalabilità e la disponibilità e possano connettersi a qualsiasi gestore code all'interno del cluster uniforme. In questo modo si rimuove qualsiasi dipendenza da un gestore code specifico, migliorando la disponibilità e il bilanciamento del carico di lavoro del traffico di messaggistica.  I cluster uniformi non sono disponibili su IBM MQ for z/OS; i gruppi di condivisione code forniscono molte delle funzioni di un cluster uniforme.

I cluster uniformi sono un modello specifico di cluster IBM MQ che fornisce una piccola raccolta di gestori code ad alta disponibilità e scalabile orizzontalmente. Questi gestori code sono configurati in modo quasi identico, in modo che un'applicazione possa interagire con essi come un singolo gruppo. Ciò rende più semplice garantire che ogni gestore code nel cluster venga utilizzato, assicurando automaticamente che le istanze dell'applicazione siano distribuite in modo uniforme tra i gestori code.

I cluster uniformi rimuovono alcune delle operazioni manuali che un amministratore deve eseguire per creare e gestire un gruppo di gestori code indipendenti e interconnessi. Spostano alcune logiche di connessione client dal client al gestore code, in cui le informazioni sui livelli di attività dell'applicazione possono informare le decisioni sui client, in merito ai gestori code a cui connettersi.

È possibile semplificare la creazione iniziale di un cluster uniforme e, successivamente, mantenere la configurazione tra i membri del cluster uniforme identica, utilizzando la configurazione automatica e il supporto del cluster automatico. Quando si utilizza questa funzionalità, un file di configurazione descrive il raggruppamento e un altro rappresenta la configurazione MQSC da applicare a tutti i gestori code nel raggruppamento uniforme. A ogni riavvio del gestore code, la configurazione viene riapplicata e il cluster si forma automaticamente. Per ulteriori dettagli sull'utilizzo di questa funzione, consultare [“Creazione di un cluster uniforme”](#) a pagina 441 .

Per trarre il massimo vantaggio da un cluster uniforme, ogni applicazione deve essere ridimensionata in più istanze corrispondenti, preferibilmente con almeno tante istanze quante sono i gestori code, se non molte di più.

Un cluster IBM MQ , di qualsiasi dimensione, fornisce più funzionalità:

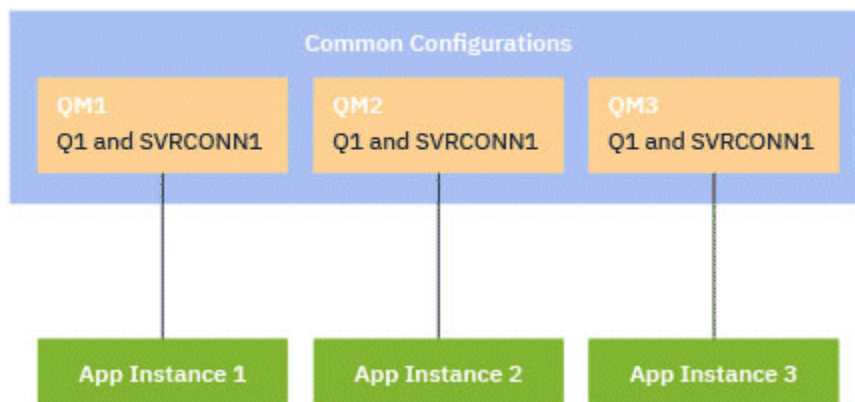
- Una directory di tutte le risorse di cluster, rilevabile da qualsiasi membro in un cluster
- Creazione automatica del canale e connettività
- Scalabilità orizzontale tra più code corrispondenti, utilizzando il bilanciamento del carico di lavoro dei messaggi
- Instradamento dinamico dei messaggi, in base alla disponibilità

I cluster uniformi utilizzano il clustering IBM MQ per la comunicazione tra gestori code e il bilanciamento del carico di lavoro tra code. Tuttavia, differiscono dai tipici cluster IBM MQ nei modi seguenti:

- I cluster uniformi in genere hanno un minor numero di gestori code nel cluster. Non creare un cluster uniforme con più di 10 gestori code.
- Ogni membro del cluster ha una configurazione quasi identica.
- Il cluster viene generalmente utilizzato da una singola applicazione o da un gruppo di applicazioni correlate.
- Il numero di istanze dell'applicazione che si collegano al cluster deve essere maggiore o uguale al numero di gestori code.

In un modello cluster uniforme, tutti i gestori code nel cluster offrono gli stessi servizi di messaggistica. Ad esempio, è possibile configurare tutti i membri del cluster in modo che abbiano le stesse code locali definite e consentire alle applicazioni client di connettersi a qualsiasi membro del cluster. È inoltre possibile che siano definiti gli stessi canali di connessione del server e, possibilmente, gli stessi record di autorizzazione, le stesse regole di autenticazione del canale e così via. Tuttavia, i membri del cluster possono ancora avere alcune differenze negli oggetti e nella configurazione. Ad esempio, alcune applicazioni potrebbero creare code dinamiche temporanee mentre sono connesse a un gestore code. Inoltre, alcuni aggiornamenti di configurazione potrebbero essere sottoposti a rollout tra i membri in un periodo di tempo; ad esempio, certificati nuovi o aggiornati. Come con i cluster IBM MQ regolari, due gestori code richiederanno una configurazione aggiuntiva per renderli gestori code del repository completi.

Il seguente diagramma mostra che i gestori code hanno configurazioni simili. Definiscono la stessa coda denominata Q1 e lo stesso canale di connessione server SVRCONN1.



Tenere presente che, per più gestori code con nomi canale di connessione server identici da utilizzare con una singola CCDT (client - channel definition table), è necessario utilizzare il formato CCDT aggiornato introdotto in IBM MQ 9.1.2. Consultare [“Configurazione di un formato JSON CCDT”](#) a pagina 46.

Nomi applicazione e istanze applicazione

Un nome applicazione viene visualizzato come attributo `APPLTAG` del comando **DISPLAY CONN(*) TYPE CONN**. Da IBM MQ 9.1.2, viene modificato il modo in cui è impostato il nome dell'applicazione.

Un'istanza di un'applicazione è un insieme di connessioni strettamente correlate che forniscono una *unità di esecuzione* per tale applicazione. In genere, questo è un singolo processo del sistema operativo, che può avere un numero di thread e connessioni IBM MQ associate.

Per ulteriori informazioni sul nome dell'applicazione e sulle istanze dell'applicazione, vedi [Concetti di sviluppo dell'applicazione](#).

Client riconnettibili

I client ricollegabili possono essere spostati per ottenere una distribuzione del carico di lavoro uniforme mentre, per definizione, un client non ricollegabile non può essere riconnesso a un gestore code diverso. Tuttavia, potrebbe esserci ancora un buon motivo per connettere un client non ricollegabile in un cluster uniforme: ad esempio, perché il client crea una forma di stato persistente e un altro meccanismo viene utilizzato per garantire che vi siano istanze dell'applicazione in esecuzione in ciascuno dei gestori code.

Applicazioni associate localmente

Si prevede che i cluster uniformi abbiano applicazioni IBM MQ che si collegano come applicazioni client, piuttosto che applicazioni associate localmente. Alle applicazioni collegate localmente non viene impedito di connettersi ai membri del cluster uniforme, ma i cluster uniformi non possono raggiungere nemmeno la distribuzione del carico di lavoro con le applicazioni associate localmente, perché non possono connettersi a nessun altro membro del cluster.

Attività correlate

[Specifica del nome dell'applicazione nei linguaggi di programmazione supportati](#)

Bilanciamento automatico dell'applicazione

Il bilanciamento automatico delle applicazioni migliora notevolmente la distribuzione e la disponibilità delle applicazioni, consentendo a un cluster uniforme IBM MQ di gestire da vicino la distribuzione delle applicazioni nel cluster, piuttosto che affidarsi alla randomizzazione o al blocco manuale delle applicazioni a specifici gestori code.

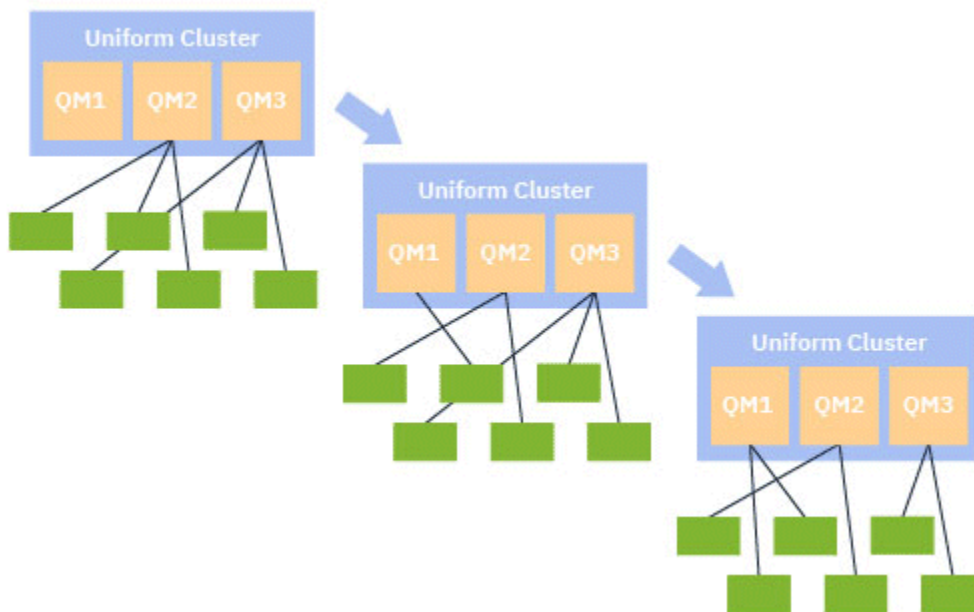
Il bilanciamento automatico tra una serie di gestori code cluster è supportato per applicazioni scritte in C, JMS, IBM MQ .NET, XMS .NET.

Quando ci sono almeno tante istanze della stessa applicazione quanti sono i gestori code, il cluster uniforme garantisce costantemente che ogni gestore code abbia almeno un'istanza dell'applicazione connessa.

Le applicazioni possono rimuovere una specifica affinità con un gestore code e utilizzare invece una CCDT (client channel definition table) per rendere casuale la connettività al gruppo di gestori code nel cluster uniforme in modo sicuro. Le applicazioni possono eseguire questa operazione per i motivi riportati di seguito:

- Quando ci sono abbastanza istanze dell'applicazione che utilizzano, c'è sempre un'istanza dei messaggi di elaborazione dell'applicazione.
- Quando si arresta un gestore code, tutte le istanze dell'applicazione connesse vengono distribuite in modo uniforme tra i gestori code rimanenti nel cluster.
- Quando si avvia un gestore code, tutte le istanze dell'applicazione connesse ad altri gestori code nel cluster vengono automaticamente ribilanciate per includere il gestore code appena avviato.

Ciò significa che il cluster uniforme garantisce continuamente la distribuzione ottimale delle applicazioni, massimizzando l'elaborazione dei messaggi, anche in caso di interruzioni pianificate e non pianificate.



Per ottenere il bilanciamento automatico, i gestori code nel cluster uniforme condividono periodicamente le informazioni tra loro. Lo fanno pubblicando i metadati sugli argomenti di sistema nel ramo \$SYS/MQ riservato della struttura ad albero degli argomenti. Ogni gestore code nel cluster uniforme sottoscrive i messaggi pubblicati da altri gestori code e crea un'immagine dello stato delle applicazioni nel cluster uniforme.

I gestori code controllano la distribuzione delle applicazioni client nell'intero cluster. Quando il numero di applicazioni connesse a un determinato gestore code è sufficientemente basso da determinare che il cluster è non bilanciato, tale gestore code pubblica una richiesta su un argomento di sistema in un altro gestore code del cluster.

Quando il messaggio viene ricevuto, il gestore code di destinazione richiede una delle applicazioni client per il reindirizzamento al gestore code richiedente. L'applicazione client riceve la richiesta di reindirizzamento, chiude la connessione e si riconnette al gestore code richiedente. Questo meccanismo di bilanciamento automatico è trasparente per l'applicazione. Per ulteriori informazioni, consultare [“Funzionamento del bilanciamento automatico”](#) a pagina 430.

Distribuendo periodicamente i meta - dati sulle applicazioni connesse, il cluster uniforme può raggiungere un rapporto ampiamente bilanciato tra le applicazioni client e i gestori code nel tempo. Per evitare che si verifichino eventi di reindirizzamento rapidi e successivi, l'algoritmo di bilanciamento automatico limita la frequenza con cui vengono effettuate richieste di reindirizzamento.

È possibile monitorare lo stato corrente delle applicazioni tra i gestori code in un cluster e monitorare istanze dell'applicazione. Per ulteriori informazioni, consultare [Monitoraggio del bilanciamento dell'applicazione](#). Puoi inoltre risolvere diversi problemi con il bilanciamento dell'applicazione, come descritto in [Risoluzione dei problemi di bilanciamento dell'applicazione](#).

Il ribilanciamento è utile solo per le applicazioni con un tempo di connessione lungo. Se si dispone di applicazioni client con tempi di connessione brevi, ad esempio le applicazioni client scritte per connettersi e disconnettersi regolarmente a gestori code differenti, è necessario configurarle come non riconnettibili. Ciò li rimuove dalla serie di applicazioni che i gestori code cercano di bilanciare.

Concetti correlati

[“Come il bilanciamento automatico utilizza la riconnessione automatica”](#) a pagina 432

Il bilanciamento automatico del cluster uniforme utilizza i miglioramenti alla funzione di riconnessione automatica esistente di IBM MQ.

Funzionamento del bilanciamento automatico

Nel cluster uniforme, le connessioni client vengono raggruppati in base al nome applicazione. Le applicazioni che si collegano a un membro del cluster uniforme utilizzando lo stesso nome applicazione sono considerate equivalenti a tutte le altre applicazioni che utilizzano lo stesso nome applicazione.

Il bilanciamento automatico garantisce una diffusione uniforme delle istanze dell'applicazione tra i membri del cluster; per ulteriori informazioni, consultare [“Nomi applicazione e istanze applicazione” a pagina 427](#). Utilizzare il comando `DISPLAY APSTATUS` per visualizzare lo stato di una o più applicazioni e istanze dell'applicazione, connesse a un gestore code o a un cluster uniforme.

Ad esempio, è possibile impostare tutte le istanze di un'applicazione di richiesta di assicurazione in modo che abbiano un nome applicazione "INSURANCE.REQUESTS". Le connessioni correlate da questa applicazione verranno automaticamente raggruppate in istanze, dove appropriato, con tutto il bilanciamento eseguito su una base per istanza.

Quando le nuove istanze dell'applicazione si collegano a un membro del cluster uniforme, l'algoritmo di bilanciamento automatico valuta quali gestori code hanno il numero minore di istanze di INSURANCE.REQUESTS e reindirizza alcune connessioni a tali gestori code.

Il bilanciamento automatico è abilitato solo nelle circostanze seguenti:

- Il valore SHARECNV del canale è maggiore di zero.
- Si verifica una delle seguenti condizioni:
 - L'applicazione client specifica MQCNO_RECONNECT
 - Il file `mqc1ient.ini` specifica **Defrecon=YES**

Nota: Le applicazioni con affinità del gestore code, ad esempio a causa di una sottoscrizione durevole o di una risposta dinamica alla coda, non possono essere ribilanciate in modo sicuro e devono utilizzare MQCNO_RECONNECT_QMGR o nessuna opzione di riconnessione.

Quando un client viene reindirizzato a un gestore code alternativo, come al solito, utilizzerà le CCDT (local client channel definition tables) per individuare le informazioni di connessione per la nuova destinazione. È quindi importante per un funzionamento regolare ed efficiente del bilanciamento automatico, che i client utilizzino una CCDT contenente una voce per ciascun membro del cluster uniforme, nonché qualsiasi gruppo di gestori code utilizzato per bilanciare le connessioni iniziali.

L'utilizzo di un formato JSON CCDT semplifica questa operazione, poiché consente più connessioni che utilizzano lo stesso nome di connessione server. Per ulteriori informazioni, consultare [“Configurazione di un formato JSON CCDT” a pagina 46](#).

Concetti correlati

[“Come il bilanciamento automatico utilizza la riconnessione automatica” a pagina 432](#)

Il bilanciamento automatico del cluster uniforme utilizza i miglioramenti alla funzione di riconnessione automatica esistente di IBM MQ.

Bilanciamento automatico delle applicazioni JMS

Quando le applicazioni Jakarta Messaging 3.0 o Java Message Service 2.0 vengono bilanciate automaticamente, i gruppi sottostanti di connessioni IBM MQ creati dalle applicazioni JMS vengono spostati insieme.

Da IBM MQ 9.3.0, la proprietà **dynamicallyBalanced** è disponibile quando si configura `ActivationSpecs`. Questa proprietà specifica se è possibile richiedere a un MDB di ricevere messaggi da un gestore code differente come parte del bilanciamento dell'applicazione in un cluster uniforme. Per ulteriori informazioni, consultare [Configurazione dell'adattatore risorse per la comunicazione in entrata](#).

Per la gestione delle connessioni JMS, i cluster uniformi hanno il concetto di *istanza dell'applicazione*. Per JMS, un' *istanza dell'applicazione* è definita come JMS Connection e tutte le relative sessioni JMS associate.

Una tag di connessione univoca viene assegnata alla connessione client che corrisponde a JMS Connection e la stessa tag viene applicata alle connessioni client che corrispondono a JMS Sessioni create da tale JMS Connection.

Ad esempio, se una coppia di applicazioni client sta eseguendo applicazioni JMS su un cluster uniforme con un singolo gestore code attivo (Gestore code 1):

- Il Client 1 crea una factory di connessione su cui imposta un nome applicazione "App1" e crea una connessione JMS e tre sessioni JMS . Il Client 1 crea quattro connessioni client nel Gestore code 1, ognuna delle quali condivide lo stesso tag di connessione, e questo viene considerato come una singola istanza diApp1".
- Il Client 2 crea anche una factory di connessione su cui imposta un nome applicazione "App1" e crea una connessione JMS e due sessioni JMS . Client 2 crea tre connessioni client ciascuna che condividono la stessa tag di connessione (distinta da quella assegnata al Client 1), e questa viene considerata come una singola istanza separata di "App1".
- Quindi, il gestore code visualizza due istanze diApp1"

Quando viene eseguito il bilanciamento automatico, le istanze dell'applicazione vengono spostate. Un gestore code sceglie un'istanza dell'applicazione (un gruppo di connessioni client che condividono la stessa tag di connessione) e richiede che l'istanza venga spostata su un gestore code diverso. Il codice client riceve la richiesta e garantisce che tutte le connessioni correlate (corrispondenti a JMS Connection e alle relative sessioni JMS associate) vengano spostate nel nuovo gestore code.

Ad esempio, prendere la serie di istanze dell'applicazione precedentemente delineata e presumere che un nuovo gestore code (Gestore code 2) venga avviato nel cluster uniforme.

Il gestore code 2 non ha lavoro ma il gestore code 1 ha 2 istanze di "App1", quindi il gestore code 2 richiede che il gestore code 1 trasferisca un'istanza di "App1" al gestore code 2.

Il gestore code 1 sceglie un'istanza diApp1" da spostare. Ai fini dell'esempio, si supponga di scegliere l'istanza creata dal client 1.

- Il Gestore code 1 invia una richiesta al Client 1 per spostare la propria istanza diApp1" in QM2.
- Il client chiude le quattro connessioni client esistenti al gestore code 1 e crea quattro nuove connessioni al gestore code 2.
- JMS Connection e le relative sessioni JMS , ad eccezione di una breve pausa nell'elaborazione, normalmente non devono essere disturbati.

Nota:

Un'applicazione potrebbe ricevere un'eccezione JMS se alcune operazioni sono in corso nel momento in cui un'istanza dell'applicazione viene spostata.

L'eccezione JMS avrà un'eccezione IBM MQ collegata, da cui è possibile richiamare il codice motivo per determinare la causa dell'errore.

I codici di errore previsti sono i seguenti:

MQRC_CALL_INTERROTTO

Ciò si verifica quando, ad esempio, un messaggio persistente (il valore predefinito in JMS) viene inserito al di fuori di un punto di sincronizzazione, ma l'operazione viene interrotta da una riconnessione.

MQRC_BACK_OUT

Ciò si verifica quando, ad esempio, un tentativo di inserire un messaggio in un punto di sincronizzazione viene interrotto da una riconnessione.

Concetti correlati

[“Funzionamento del bilanciamento automatico” a pagina 430](#)

Nel cluster uniforme, le connessioni client vengono raggruppati in base al nome applicazione. Le applicazioni che si collegano a un membro del cluster uniforme utilizzando lo stesso nome applicazione sono considerate equivalenti a tutte le altre applicazioni che utilizzano lo stesso nome applicazione.

[“Come il bilanciamento automatico utilizza la riconnessione automatica” a pagina 432](#)

Il bilanciamento automatico del cluster uniforme utilizza i miglioramenti alla funzione di riconnessione automatica esistente di IBM MQ.

Multi

Come il bilanciamento automatico utilizza la riconnessione automatica

Il bilanciamento automatico del cluster uniforme utilizza i miglioramenti alla funzione di riconnessione automatica esistente di IBM MQ.

Nelle versioni di IBM MQ precedenti a IBM MQ 9.2.0, la funzione di riconnessione automatica si riconnette automaticamente a un'istanza in standby di un gestore code o a un gestore code differente, in base ai dettagli di connessione forniti, generalmente un elenco di nomi di connessione o una CCDT (Client Channel Definition Table).

In alcune circostanze, il client IBM MQ esegue in modalità non presidiata la riconnessione senza che l'applicazione sia a conoscenza del fatto che ciò si è verificato. La decisione del gestore code a cui riconnettersi è interamente legata alla sequenza di nomi di connessione in un elenco di nomi di connessione o alla configurazione del bilanciamento del carico di lavoro in CCDT.

Da IBM MQ 9.2.0 è possibile inviare una richiesta di riconnessione a un client contenente un suggerimento sul gestore code a cui il client deve riconnettersi. In molti scenari di riconnessione, come ad esempio un errore del gestore code o l'amministratore che immette il comando **endmqm -r**, un nome del gestore code non è incluso nelle informazioni di suggerimento e il funzionamento della riconnessione automatica funziona come prima di IBM MQ 9.2.0.

Tuttavia, se è stato configurato un cluster uniforme, il bilanciamento automatico dell'applicazione invia periodicamente le richieste di riconnessione ai client, in modo da ottenere un cluster bilanciato. In questi casi, il cluster uniforme specifica un nome gestore code nel suggerimento di riconnessione per garantire che le connessioni client vengano spostate sui gestori code che hanno il numero minimo di connessioni.

Perché il bilanciamento automatico funzioni, devono essere presenti le seguenti impostazioni:

- Le applicazioni IBM MQ utilizzano i CCDT per richiamare le informazioni di connessione.
- I CCDT contengono una voce per ogni gestore code nel cluster uniforme.

In caso contrario, il cluster non può bilanciare automaticamente le applicazioni tra tutti i membri del cluster.

Se un'applicazione sta utilizzando una versione del client IBM MQ precedente a IBM MQ 9.2.0 ed è configurata per supportare la riconnessione automatica del client, potrebbe essere inviata una richiesta dal cluster uniforme per passare attraverso i suoi passi di riconnessione.

Al client non verrà richiesto di riconnettersi a uno specifico gestore code, ma verrà eseguita la stessa sequenza di logica di riconnessione che avrebbe per altri eventi di riconnessione. Per ottenere una distribuzione uniforme delle applicazioni client prima di IBM MQ 9.2.0 nel cluster uniforme, assicurarsi che i client siano configurati per utilizzare CCDT contenenti voci con peso uniforme per ogni membro del cluster.

Le applicazioni possono effettuare diversi tentativi di riconnessione prima di connettersi a un gestore code che necessita di un'istanza aggiuntiva, quindi è un metodo meno efficiente per ottenere una distribuzione uniforme delle applicazioni nel cluster. In questi ambienti, il bilanciamento automatico può richiedere più tempo.

I client IBM MQ non supportano la riconnessione automatica del client

Se un'applicazione sta utilizzando una versione del client IBM MQ che non supporta la riconnessione automatica del client, l'applicazione potrebbe ricevere un codice di ritorno di errore da una chiamata MQI.

Se l'applicazione non è stata progettata per gestire gli errori ed eseguire le riconnessioni manualmente, potrebbe essere necessario disabilitare il bilanciamento automatico per tali applicazioni.

Nota: Il bilanciamento automatico è abilitato per qualsiasi applicazione identificata come ricollegabile, ossia, l'applicazione ha MQCNO_RECONNECT nelle sue opzioni di connessione effettive.

Attività correlate

[“Creazione di un nuovo cluster uniforme” a pagina 442](#)

Come creare un nuovo cluster uniforme.

Multi *Influenza del ribilanciamento delle applicazioni in cluster uniformi*

Con il bilanciamento automatico dell'applicazione (una funzione di cluster uniformi), a una connessione dell'applicazione può essere richiesto di passare a un gestore code alternativo in qualsiasi punto del suo ciclo di vita.

Introduzione

Da IBM MQ 9.3.0, l'algoritmo di bilanciamento tenta automaticamente di prendere in conto lo stato delle applicazioni per ridurre al minimo l'interruzione del flusso dell'applicazione. Questo può essere regolato per adattarsi a particolari applicazioni o istanze dell'applicazione fornendo a IBM MQ ulteriori informazioni sul tipo di applicazione o sul modello dell'attività IBM MQ eseguita da questa applicazione.

Di solito, la persona che sviluppa o distribuisce un'applicazione client si trova nella posizione migliore per comprendere questo modello e fornire queste informazioni al gestore code (consultare [Distribuzione di applicazioni client flessibili e scalabili](#)), ma potrebbe anche, o in aggiunta, essere regolata da un responsabile.

Tenere presente che se il gestore code non è in grado di ottenere una distribuzione uniforme delle applicazioni entro un periodo di tempo ragionevole, le connessioni dell'applicazione potrebbero ancora essere ribilanciate ad altri gestori code senza attendere un periodo di tempo appropriato nel loro flusso IBM MQ .

Questo può anche essere regolato per soddisfare i requisiti. Se è più importante raggiungere rapidamente una distribuzione uniforme delle applicazioni, è possibile configurare il prodotto in modo da attendere meno tempo per trovare un tempo adatto per ribilanciare un'applicazione. In alternativa, se è più importante impedire l'interruzione delle applicazioni, è possibile configurare il prodotto in modo che attenda sempre un tempo conveniente per spostare l'applicazione.

Vedi [Distribuzione di applicazioni client flessibili e scalabili](#) per ulteriori informazioni di panoramica.

Per le applicazioni .NET, consultare [“Influenza del ribilanciamento dell'applicazione in .NET” a pagina 436](#) per ulteriori informazioni.

Per.XMS.NET , consultare [Proprietà di ConnectionFactory](#) per ulteriori informazioni.

V 9.4.0 Per le applicazioni JMS, consultare [“Influenza del ribilanciamento dell'applicazione in IBM MQ classes for JMS” a pagina 437](#) per ulteriori informazioni.

Funzionamento del bilanciamento dell'applicazione predefinito

Per impostazione predefinita, lo stato transazione / unità di lavoro di un'interazione delle applicazioni con un gestore code viene considerato per tutte le applicazioni.

Per le transazioni locali, il bilanciamento automatico delle applicazioni evita l'emissione di richieste di ribilanciamento alle applicazioni attualmente coinvolte in una transazione. Sebbene ciò non elimini la possibilità che un'applicazione riceva un codice di ritorno di cui è stato eseguito il backout, poiché il raggiungimento del timeout di ribilanciamento configurato o un'interruzione effettiva potrebbe ancora causare un tale codice di ritorno, significa che alle applicazioni di solito non verrà richiesto di riconnettersi mentre si trovano nel mezzo di una transazione.

Per le applicazioni che iniziano una nuova transazione quasi immediatamente dopo aver completato quella precedente, potrebbe verificarsi un ritardo per la chiamata iniziale nella nuova transazione durante il completamento del ribilanciamento. Ciò garantisce che il bilanciamento automatico delle applicazioni sia ancora in grado di raggiungere una distribuzione uniforme delle applicazioni tra gestori code in un cluster uniforme.

Se si dispone di applicazioni che utilizzano transazioni con esecuzione più lunga, è possibile aumentare il valore del timeout di ribilanciamento o disabilitare completamente questo vincolo. Consultare

“Configurazione del comportamento di bilanciamento” a pagina 434 per i link su come controllarlo in MQI e .NET o 'Progettazione delle applicazioni client per la tolleranza degli errori e la scalabilità' per l'equivalente del livello di codice.

Bilanciamento richiesta - risposta

Quando il tipo di applicazione è specificato come **Request-Reply**, è prevista una risposta GET per ogni PUT eseguito dall'istanza dell'applicazione. Se l'istanza dell'applicazione coinvolge più thread o gestisce richieste e risposte in batch, più richieste e risposte possono essere in esecuzione in qualsiasi momento.

L'applicazione non è considerata idonea per lo spostamento, fino a quando il numero di richieste inviate non è uguale al numero di risposte ricevute o viene superato il valore di backstop del timeout.

Un'eccezione si verifica quando la scadenza del messaggio è configurata per un messaggio di richiesta. Si presume che le risposte debbano essere ricevute entro l'intervallo di scadenza del messaggio di richiesta e quando tutti i messaggi di richiesta sono scaduti, l'algoritmo di bilanciamento non attende più ulteriori risposte prima di considerare l'istanza idonea per lo spostamento.

Se più richieste sono in attesa, viene considerata solo la scadenza più recente tra i messaggi di richiesta inviati. Quando si utilizzano valori di scadenza significativi, è necessario configurare il parametro di bilanciamento **Timeout** in modo che l'applicazione abbia un valore almeno pari a quello di qualsiasi scadenza del messaggio inviato, per evitare di abbreviare la finestra di scadenza della richiesta/risposta prevista.

Il modello precedente è adatto solo per applicazioni che prevedono periodi in cui non vi sono richieste in sospenso. Applicazioni complesse a più thread, che inviano e ricevono costantemente messaggi, ad esempio, potrebbero non diventare mai idonee per il ribilanciamento in base a questo modello.

Note:

- Non viene effettuato alcun tentativo di correlare richieste e risposte specifiche, pertanto, se una risposta precedente all'interno di un batch di messaggi in corso scade, l'applicazione potrebbe ancora attendere la scadenza dell'ultima richiesta prima di essere idonea per il bilanciamento.
- In particolare, è necessario prestare attenzione se si combinano un tempo di scadenza illimitato e messaggi in scadenza, per motivi simili.

Se i messaggi di richiesta con una scadenza limitata sono in sospenso e i nuovi messaggi vengono inviati con una scadenza illimitata, il timeout di scadenza illimitata *non* viene preso in considerazione dall'algoritmo di bilanciamento, che continua a rispettare l'ultima ora di scadenza corrente.

In caso contrario, le risposte precedenti potrebbero impedire alla domanda di essere eleggibili per lo spostamento. Di conseguenza, se le risposte di timeout di scadenza illimitate sono in attesa, ma le richieste in scadenza vengono successivamente inviate, il tempo di attesa viene ridotto alla scadenza più lunga (limitata).

In generale, è necessario evitare che una singola istanza dell'applicazione invii messaggi di richiesta in scadenza e non in scadenza in un'applicazione bilanciata, poiché l'eleggibilità al ribilanciamento diventa più difficile per uno sviluppatore o un amministratore tracciare o definire in modo accurato.

- Solo il tempo di scadenza specificato dall'applicazione mittente (ad esempio, in MQI il valore di MQMD.**Expiry**) viene considerato quando si determina il tempo di attesa per le risposte. Le successive modifiche a questo valore, ad esempio, l'utilizzo di CAPEXPY non influirà sul tempo di attesa.

Configurazione del comportamento di bilanciamento

Per influenzare precisamente quando IBM MQ ribilancia le applicazioni, alcuni ambienti di applicazioni client possono fornire informazioni in fase di connessione sul modello di messaggistica utilizzato.

Queste informazioni sono fornite in una nuova struttura denominata *Opzioni di bilanciamento*.

Per MQI, consultare “Configurazione del comportamento di bilanciamento utilizzando MQI” a pagina 435.

Per l'equivalente client .NET di questa struttura, consultare [“Influenza del ribilanciamento dell'applicazione in .NET”](#) a pagina 436.

V 9.4.0 Per l'approccio JMS all'impostazione di queste opzioni, consultare [“Influenza del ribilanciamento dell'applicazione in IBM MQ classes for JMS”](#) a pagina 437 per ulteriori informazioni.

Altri ambienti client attualmente non supportano la fornitura di questa struttura al momento della connessione.

Multi *Configurazione del comportamento di bilanciamento utilizzando MQI*

Per influenzare precisamente quando IBM MQ ribilancia le applicazioni, alcuni ambienti di applicazioni client possono fornire informazioni in fase di connessione sul modello di messaggistica utilizzato.

In MQI, la struttura delle opzioni di bilanciamento viene definita [MQBNO](#).

Se nel programma non vengono fornite *Opzioni di bilanciamento*, i client di supporto ricavano queste informazioni nella stanza [Stanza dell'applicazione](#) o nella stanza [ApplicationDefaults](#) nel file `client.ini` distribuito insieme all'applicazione client.

Nota: Queste stanze sono identiche, ad eccezione del fatto che la versione `Application` contiene un campo **Name** per identificare l'applicazione a cui si applicano queste opzioni.

Se viene fornita una delle due forme di stanza, tutti i campi devono essere presenti tranne **BalanceOptions** che si presume sia none se non esplicitamente impostato.

L'ordine di preferenza per la fornitura delle opzioni è:

1. Una struttura MQBNO viene fornita dall'applicazione su CONNX e utilizzata nella sua interezza
2. Oppure, la stanza `Application` denominata corrispondente, se presente, viene utilizzata solo per generarne una
3. Oppure, la stanza `ApplicationDefaults`, se presente, viene utilizzata solo per generare una
4. Oppure, nessun flusso MQBNO per questa connessione.

È possibile fornire tre parti chiave di informazioni dalla struttura MQBNO o dal file `client.ini`:

1. Il **ApplicationType** o il modello dell'applicazione.

Questo campo indica IBM MQ il pattern generale di IBM MQ attività a cui partecipa questa applicazione.

Sono supportati tre tipi di applicazione:

Semplice

Nessuna regola specifica deve essere applicata al di là dei valori predefiniti descritti in [“Funzionamento del bilanciamento dell'applicazione predefinito”](#) a pagina 433.

Richiesta-Risposta

Dopo ogni chiamata MQPUT, è prevista una chiamata MQGET corrispondente per un messaggio di risposta. Consultare [“Bilanciamento richiesta - risposta”](#) a pagina 434 per maggiori dettagli.

Client gestito

Le richieste di ribilanciamento vengono sempre inviate immediatamente al client, che esegue il ribilanciamento in un punto che considera appropriato, ad esempio, l'adattatore di risorse JEE viene registrato in questo modo.

2. Il **Timeout** dopo il quale il ribilanciamento può interrompere l'attività dell'applicazione
3. Specifica **BalanceOptions**

Esempi di quando la tua applicazione potrebbe essere ribilanciata

Esempio 1

È stata scritta un'applicazione che inserisce i messaggi nel punto di sincronizzazione ed esegue il commit del batch di messaggi emettendo una chiamata MQCMIT. Quando la chiamata MQCMIT viene completata, l'applicazione inizia a inserire i messaggi in un nuovo punto di sincronizzazione.

Configurazione IBM MQ suggerita

Opzioni predefinite sufficienti

Risultato

Un'istanza dell'applicazione viene spostata dopo che una chiamata MQCMIT ha avuto esito positivo (o negativo), una volta raggiunto il numero configurato di transazioni.

Per impostazione predefinita, se un batch di messaggi supera i 10 secondi, è possibile che venga eseguito il rollback se è stato richiesto un ribilanciamento. Se si prevede che le transazioni superino regolarmente questo limite e che ciò sia consentito, è possibile estendere il **Timeout** in modo appropriato.

Esempio 2

Dopo l'elaborazione della richiesta, è stata scritta un'applicazione che inserisce un messaggio in un'istanza della coda cluster e un'altra applicazione risponde a una coda dinamica temporanea locale con un messaggio. Quando la richiesta è stata letta in modo distruttivo dalla coda locale, l'applicazione inserisce il messaggio di richiesta successivo.

Configurazione IBM MQ suggerita

Impostare Tipo su MQBNO_BALTYPE_REQREP

Risultato

Un'istanza dell'applicazione viene spostata quando un'applicazione completa una chiamata MQGET, a questo punto l'istanza dell'applicazione si sposta su un altro gestore code. Tutte le chiamate MQPUT successive vengono eseguite sul nuovo gestore code.

MQBNO

ApplicationType

Multi *Influenza del ribilanciamento dell'applicazione in .NET*

Da IBM MQ 9.3.0, sono disponibili ulteriori costanti per impostare le proprietà dell'opzione di bilanciamento utilizzando una tabella hash dall'applicazione quando viene utilizzata la classe `MQQueueManager` per la connessione al gestore code.

Le seguenti costanti sono quelle utilizzate per influenzare il bilanciamento dell'applicazione in .NET:

Ribilanciamento del tipo di applicazione

Il tipo di azione di bilanciamento; rappresentato dalla costante

MQC.BALANCING_APPLICATION_TYPE_PROPERTY

- È necessario utilizzare questa proprietà per impostare il campo **ApplicationType** della struttura MQBNO.

È necessario impostare i valori di tipo intero e i valori possibili sono:

MQC.BALANCING_APPLICATION_TYPE_SIMPLE

Bilanciamento semplice; non vengono applicate regole specifiche oltre a quelle descritte in [“Influenza del ribilanciamento delle applicazioni in cluster uniformi”](#) a pagina 433. Questo è il valore predefinito.

MQC.BALANCING_APPLICATION_TYPE_REQUEST_REPLY

Bilanciamento richiesta - risposta; dopo ogni chiamata MQPUT, è prevista una chiamata MQGET corrispondente per un messaggio di risposta. Il bilanciamento viene ritardato fino a quando non viene ricevuto un messaggio di questo tipo o fino a quando non viene superato il messaggio di richiesta **EXPIRY**.

Se la riconnessione è abilitata dall'applicazione e questa proprietà non è impostata, viene utilizzato **MQC.BALANCING_APPLICATION_TYPE_SIMPLE**

Opzioni di ribilanciamento

Le opzioni di bilanciamento impostate dall'applicazione emittente; rappresentate dalla costante **MQC.BALANCING_OPTIONS_PROPERTY**

- È necessario utilizzare questa proprietà per impostare il campo **BalanceOptions** della struttura MQBNO.

È necessario impostare i valori di tipo intero e i valori possibili sono:

MQC.BALANCING_OPTIONS_NONE

Nessuna opzione impostata. Questo è il valore predefinito

MQC.BALANCING_OPTIONS_IGNORE_TRANSACTIONS

L'impostazione di questa opzione consente alle applicazioni di essere ribilanciate anche se nel mezzo di una transazione.

Se la riconnessione è abilitata dall'applicazione e questa proprietà non è impostata, viene utilizzato **MQC.BALANCING_OPTIONS_NONE**.

Timeout di ribilanciamento

Timeout dopo il quale il ribilanciamento potrebbe interrompere l'attività dell'applicazione; rappresentato dalla costante **MQC.BALANCING_TIMEOUT_PROPERTY**

- È necessario utilizzare questa proprietà per impostare il campo **Timeout** della struttura MQBNO.

È necessario impostare i valori di tipo intero e i valori possibili sono:

MQC.BALANCING_TIMEOUT_AS_DEFAULT

Il valore di timeout predefinito impostato. Questo è il valore predefinito

MQC.BALANCING_TIMEOUT_IMMEDIATE

Si verifica un timeout immediato

MQC.BALANCING_TIMEOUT_NEVER

Non si verifica alcun timeout

Nota: È necessario fornire un solo valore dai valori definiti oppure un valore compreso tra 0 e 999999999 secondi.

Distribuzione di applicazioni client flessibili e scalabili

MQBNO

V 9.4.0 **Multi** *Influenza del ribilanciamento dell'applicazione in IBM MQ classes for JMS*

Da IBM MQ 9.4.0, sono disponibili ulteriori costanti per impostare le proprietà dell'opzione di bilanciamento su un **ConnectionFactory**. Queste costanti sono applicabili solo se **WMQ_PROVIDER_VERSION** è impostato su 7. Le applicazioni Request_reply in un cluster uniforme devono consentire la possibilità di risposte mancanti.

- [“Le costanti disponibili” a pagina 437.](#)
- [“Il potenziale per i messaggi persi durante il bilanciamento delle applicazioni REQUEST_REPLY” a pagina 438.](#)

Le costanti disponibili

Le seguenti costanti sono quelle utilizzate per influenzare il bilanciamento dell'applicazione in IBM MQ classes for JMS:

Ribilanciamento del tipo di applicazione

Il tipo di azione di bilanciamento; rappresentato dalla costante **WMQConstants.WMQ_BALANCING_APPLICATION_TYPE**

- È necessario utilizzare questa proprietà per impostare il campo **ApplicationType** della struttura MQBNO.

È necessario impostare valori di tipo intero. I valori possibili sono i seguenti:

WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_SIMPLE (impostazione predefinita)

Bilanciamento semplice; non vengono applicate regole specifiche oltre a quelle descritte in [“Influenza del ribilanciamento delle applicazioni in cluster uniformi”](#) a pagina 433.

WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_REQUEST_REPLY

Bilanciamento richiesta - risposta; dopo ogni chiamata **MQPUT**, è prevista una chiamata **MQGET** corrispondente per un messaggio di risposta. Il bilanciamento viene ritardato fino a quando non viene ricevuto un messaggio di questo tipo o fino a quando non viene superato il messaggio di richiesta **EXPIRY**.

Se la riconnessione è abilitata dall'applicazione e questa proprietà non è impostata, viene utilizzato **WMQConstants.WMQ_BALANCING_APPLICATION_TYPE_SIMPLE**.

Opzioni di ribilanciamento

Le opzioni di bilanciamento impostate dall'applicazione emittente; rappresentate dalla costante **WMQConstants.WMQ_BALANCING_OPTIONS**

- È necessario utilizzare questa proprietà per impostare il campo **BalanceOptions** della struttura **MQBNO**.

È necessario impostare valori di tipo intero. I valori possibili sono i seguenti:

WMQConstants.WMQ_BALANCING_OPTIONS_NONE (impostazione predefinita)

Nessuna opzione impostata.

WMQConstants.WMQ_BALANCING_OPTIONS_IGNORE_TRANSACTIONS

L'impostazione di questa opzione consente alle applicazioni di essere ribilanciate anche se nel mezzo di una transazione.

Se la riconnessione è abilitata dall'applicazione e questa proprietà non è impostata, viene utilizzato **WMQConstants.WMQ_BALANCING_OPTIONS_NONE**.

Timeout di ribilanciamento

Il timeout dopo il quale il ribilanciamento potrebbe interrompere l'attività dell'applicazione; rappresentato dalla costante **WMQConstants.WMQ_BALANCING_TIMEOUT**

- È necessario utilizzare questa proprietà per impostare il campo **Timeout** della struttura **MQBNO**.

È necessario impostare valori di tipo intero. I valori possibili sono i seguenti:

WMQConstants.WMQ_BALANCING_TIMEOUT_AS_DEFAULT (impostazione predefinita)

Il valore di timeout predefinito impostato. Per impostazione predefinita, questo valore è 10 secondi.

WMQConstants.WMQ_BALANCING_TIMEOUT_IMMEDIATE

Si verifica un timeout immediato.

WMQConstants.WMQ_BALANCING_TIMEOUT_NEVER

Non si verifica alcun timeout.

un valore compreso tra 1 e 999999999

Rappresenta un valore in secondi.

Nota: È necessario fornire un solo valore dai valori definiti oppure un valore compreso tra 0 e 999999999 secondi.

Queste proprietà possono essere impostate anche nelle rappresentazioni JNDI dei factory di connessione utilizzando le interfacce **JMSAdmin** o **IBM MQ Explorer**.

Il potenziale per i messaggi persi durante il bilanciamento delle applicazioni REQUEST_REPLY

In **IBM MQ classes for JMS** (e **IBM MQ classes for Jakarta Messaging**), la funzionalità di richiesta/risposta viene implementata impostando la proprietà **JMSReplyTo** sul messaggio di richiesta, che viene utilizzato dall'applicazione che risponde per determinare se la risposta viene inviata. In termini **JMS**, la proprietà **JMSReplyTo** è un **Destination**.

Quando questa viene convertita in operazioni IBM MQ , la proprietà **JMSReplyTo** viene inviata come URI della coda completo - che identifica una coda su uno specifico gestore code.

A causa della natura asincrona della gestione delle riconessioni di bilanciamento, una riconnessione potrebbe essere avviata dopo che la proprietà **JMSReplyTo** è stata convertita in un URI completo, ma prima che il messaggio di richiesta sia stato inserito nella coda di richieste. In queste circostanze, l'applicazione che risponde potrebbe inviare la sua risposta alla coda di risposta originale sul gestore code originale, ma l'applicazione richiedente potrebbe ora essere in attesa di una risposta sul nuovo gestore code.

Le applicazioni Request_reply in un cluster uniforme devono quindi consentire la possibilità di risposte perse.

[Distribuzione di applicazioni client flessibili e scalabili](#)
[MQBNO - Opzioni di bilanciamento](#)

Multi *Limitazioni e considerazioni per i cluster uniformi*

Limitazioni e altri punti da considerare quando si configurano cluster uniformi.

Nota: Per i requisiti generali durante la configurazione di cluster uniformi, consultare anche [“Creazione di un nuovo cluster uniforme”](#) a pagina 442.

Importanza dell'uniformità tra i gestori code

Per impostazione predefinita, qualsiasi applicazione che si dichiara come reconnectable potrebbe essere ribilanciata in un gestore code alternativo in un cluster uniforme in qualsiasi momento. Ciò significa che qualsiasi risorsa, ad esempio, coda, argomento o record di autorizzazione richiesto da tali applicazioni deve essere dichiarato su tutti i gestori code nel cluster uniforme.

La coerenza della configurazione del gestore code non è controllata. Spetta all'amministratore di sistema configurare i membri del cluster in modo che abbiano una configurazione simile.

Tuttavia, è possibile facilitare la congruenza utilizzando la funzione [Configurazione automatica da uno script MQSC all'avvio](#) per condividere gli script MQSC che definiscono gli oggetti per il cluster e, quindi, assicurarsi che tutti abbiano le stesse definizioni. Per ulteriori informazioni, consultare [“Creazione di un nuovo cluster uniforme”](#) a pagina 442.

Questa uniformità si estende ai gestori code del repository completo per il cluster. Anche se per i cluster IBM MQ tradizionali è spesso considerata la procedura ottimale per separare i repository completi in sistemi autonomi, in un cluster uniforme, il modello è che i repository completi partecipano completamente al cluster e elaborano i carichi di lavoro dell'applicazione insieme ad altri nodi.

Sovrapposizione di cluster uniformi e cluster IBM MQ tradizionali

Un gestore code del cluster uniforme può partecipare al massimo a un cluster uniforme e può anche essere membro di un qualsiasi numero di cluster IBM MQ standard. Potrebbe essere utile pensare che il cluster uniforme agisca come un singolo gestore code nel cluster più ampio.

Tenere presenti le seguenti considerazioni:

- Un gestore code del cluster uniforme che agisce come repository completo, deve essere solo un repository completo per il cluster uniforme stesso.
- Allo stesso modo, i gestori code del repository parziale che sono membri di un cluster uniforme, ma potrebbero anche appartenere a un cluster IBM MQ tradizionale più ampio, non possono essere utilizzati come repository esterno al cluster uniforme.

Per ulteriori informazioni, consultare [Come scegliere i gestori code del cluster per conservare i repository completi](#).

Il motivo è che i gestori code che sono repository completi per una combinazione di cluster IBM MQ tradizionali e cluster uniformi, incoraggiano una divergenza di dati contenuti nella cache del cluster tra i

membri del cluster uniforme e quindi si spostano rispetto all'utilizzo della funzione cluster uniforme come previsto.

Per sostituire un singolo gestore code del repository completo con un cluster uniforme, separare il repository completo dal lavoro dell'applicazione in corso su di esso e spostare solo il lavoro dell'applicazione nel cluster uniforme.

Quando si utilizzano le definizioni automatiche per i cluster uniformi, i canali del cluster non possono essere condivisi per l'utilizzo in altri cluster, ossia, si imposta l'attributo **CLUSTER** sul cluster automatico e l'attributo **CLUSNL** deve essere vuoto.

Considerazioni sul bilanciamento dell'applicazione

Le istanze dell'applicazione non sono sempre bilanciate in modo uniforme, in particolare nelle seguenti circostanze:

- Quando ci sono meno istanze dell'applicazione dei gestori code nel cluster.
- Durante un breve periodo di tempo dopo che le applicazioni client si collegano al cluster o ne escono.

Per evitare che le applicazioni client vengano ribilanciate troppo spesso, soprattutto quando le connessioni delle applicazioni sono in corso, vengono impostati dei limiti sulla frequenza con cui il cluster uniforme richiede il ribilanciamento delle applicazioni client. Dopo un periodo di alta attività di connessione o disconnessione, potrebbero essere necessari diversi minuti per bilanciare in modo uniforme le istanze dell'applicazione rimanenti nel cluster uniforme.

Per ulteriori informazioni, vedi [Risoluzione dei problemi di bilanciamento dell'applicazione](#).

Affinità applicazione

Non tutte le applicazioni sono adatte per il ribilanciamento automatico in un cluster uniforme. Vengono ribilanciati solo le applicazioni che specificano **MQCNQ_RECONNECT**. Le applicazioni che hanno un'affinità con un particolare gestore code devono specificare l'opzione **MQCNQ_NO_RECONNECT** o **MQCNQ_RECONNECT_Q_MGR**. Quest'ultimo consente il failover HA ma non il ribilanciamento.

Esempi di applicazioni che creano un'affinità ... implicita per un gestore code:

- Applicazioni che creano sottoscrizioni durevoli.
- Applicazioni che creano code dinamiche permanenti, ad esempio, per ricevere messaggi di risposta.
- Le applicazioni che prevedono un ordinamento rigoroso dei messaggi o richiedono che tutti i messaggi in una sequenza vengano elaborati dalla stessa istanza dell'applicazione o da entrambe.

Queste applicazioni devono specificare le opzioni **MQCNQ_NO_RECONNECT** o **MQCNQ_RECONNECT_Q_MGR** piuttosto che **MQCNQ_RECONNECT**.

Per ulteriori informazioni, consultare [Opzioni di riconnessione](#).

Disponibilità messaggi

Mentre il bilanciamento dell'applicazione può ribilanciare le connessioni intorno ai gestori code non riusciti o temporaneamente non disponibili, i cluster uniformi non replicano i dati dei messaggi tra i propri membri. Per la disponibilità dei dati, se un nodo ha esito negativo, anche ciascun membro del cluster uniforme deve essere configurato per essere altamente disponibile. Molte soluzioni di replica dei dati e alta disponibilità sono disponibili e possono essere combinate con cluster uniformi per la massima disponibilità di dati e servizi, ad esempio:

- L'archiviazione replicata che supporta una istanza del contenitore riavviata automaticamente dall'orchestrazione del contenitore. Per ulteriori informazioni, consultare [Single resilient queue manager](#).
- Gestori code RDQM. Per ulteriori informazioni, consultare [Alta disponibilità RDQM](#).
- Gestori code a più istanze. Per ulteriori informazioni, consultare [“Gestori code a più istanze” a pagina 524](#).

- HA nativa. Per ulteriori informazioni, vedi [HA nativa](#).
- IBM MQ Appliance HA. Per ulteriori informazioni, vedi [Elevata disponibilità](#).

Scalabilità e prestazioni dei cluster uniformi

Per consentire una maggiore integrazione e condivisione dello stato dell'applicazione tra i gestori code in un cluster uniforme, è necessario un livello più elevato di intercomunicazione rispetto a un cluster IBM MQ tradizionale. Pertanto, il ridimensionamento a un numero elevato di gestori code in un singolo cluster uniforme non è consigliato perché la comunicazione aggiuntiva ha un effetto negativo sulle prestazioni.

Sia per motivi di prestazioni che di gestione, è preferibile considerare un cluster uniforme come un singolo gestore code tradizionale che fornisce la messaggistica a un numero di applicazioni correlate, ma non è un servizio di messaggistica unico in un'azienda. In questo modello, i numeri piccoli, fino a 10, sono generalmente sufficienti per supportare un numero elevato di connessioni delle applicazioni client. Il bilanciamento dell'applicazione rende semplice iniziare con numeri piccoli, ad esempio 3 gestori code, e aumentare aggiungendo ulteriori gestori code.



Attenzione: L'abilitazione del funzionamento del cluster uniforme in un cluster che non presenta le caratteristiche consigliate, in particolare l'utilizzo di cluster con un numero elevato di gestori code, potrebbe avere un grave impatto sulle prestazioni.

Concetti correlati

[“Bilanciamento automatico dell'applicazione” a pagina 428](#)

Il bilanciamento automatico delle applicazioni migliora notevolmente la distribuzione e la disponibilità delle applicazioni, consentendo a un cluster uniforme IBM MQ di gestire da vicino la distribuzione delle applicazioni nel cluster, piuttosto che affidarsi alla randomizzazione o al blocco manuale delle applicazioni a specifici gestori code.

Creazione di un cluster uniforme

È possibile semplificare sia la creazione iniziale di un cluster uniforme, sia il mantenimento della configurazione tra i membri del cluster uniforme, utilizzando la configurazione automatica e il supporto del cluster automatico.

Prima di iniziare

Prima di creare un cluster uniforme, è necessario leggere [“Limitazioni e considerazioni per i cluster uniformi” a pagina 439](#).

Informazioni su questa attività

Si indica che un cluster IBM MQ particolare deve essere trattato come un cluster uniforme fornendo nel file `qm.ini` una sezione per AutoCluster con almeno **Type=Uniform** e **ClusterName=< uniform cluster name>**.

Puoi facoltativamente configurare il cluster IBM MQ sottostante tramite la stessa stanza `.ini` utilizzando la *creazione automatica del cluster*. Quando si usa questo supporto cluster automatico per configurare il cluster, si fornisce un file di configurazione che descrive il cluster e i relativi repository completi.

Se il gestore code in fase di avvio è elencato come uno dei repository completi, viene automaticamente reso un repository completo. Allo stesso modo, quando il canale ricevente del cluster è definito, i canali mittente del cluster per il repository completo o i repository vengono definiti automaticamente.

Procedura

Per sfruttare ulteriori funzioni che richiedono un cluster uniforme, è necessario completare una delle seguenti operazioni:

- [Convertire un cluster esistente in un cluster uniforme](#), che corrisponde al modello descritto in [“Informazioni sui cluster uniformi” a pagina 426](#).

- Creare un nuovo cluster uniforme per questo scopo.

Creazione di un nuovo cluster uniforme

Come creare un nuovo cluster uniforme.

Procedura

1. Creare un file che descrive come si desidera che sia il cluster stesso in termini di repository completi.

Come per qualsiasi cluster, due repository completi fungono da archivi centrali di informazioni sul cluster.

In particolare, è necessario descrivere i nomi e i nomi di connessione per i due repository completi in questo cluster.

Nota: Questa operazione viene eseguita in anticipo rispetto alla creazione di qualsiasi elemento (inclusi i gestori code) e il seguente processo include la creazione di tali gestori code.

Ad esempio, si supponga di configurare un cluster uniforme denominato UNICLUS, con membri del gestore code QMA, QMB, QMC e QMD. In questo esempio, QMA e QMB saranno i repository completi, con QMC e QMD come repository parziali. Un file di configurazione di esempio, `uniclus.ini`:

```
AutoCluster:
  Repository2Conname=QMA.dnsname(1414)
  Repository2Name=QMA
  Repository1Conname=QMB.dnsname(1414)
  Repository1Name=QMB
  ClusterName=UNICLUS
  Type=Uniform
```

I campi **RepositoryNConname** vengono utilizzati come attributo *conname* per altri membri del cluster per definire i mittenti del cluster (CLUSDR) e possono essere un elenco di connessioni per un gestore code a più istanze e possono includere facoltativamente la porta.

2. Creare un file di configurazione di esempio, `uniclus.mqsc`, che contenga le definizioni MQSC che si desidera applicare a tutti i membri del cluster.

C'è una riga obbligatoria necessaria in questo file, che è una definizione di un canale ricevente del cluster (CLUSRCVR), con un attributo CLUSTER del nome cluster automatico (di solito tramite + AUTOCL + insert) e un nome canale che include + QMNAME + insert.

Descrive in che modo gli altri membri del cluster uniforme si connettono a ciascun gestore code e viene utilizzato come modello per la connessione anche agli altri gestori code. Una definizione di esempio potrebbe essere:

```
define channel('+AUTOCL+_QMNAME+') chltype(clusrcvr) trdtype(tcp)
conname(+CONNAME+) cluster('+AUTOCL+') replace
```

Quando vengono configurati i cluster automatici, una definizione di un canale ricevente cluster può utilizzare alcuni inserimenti aggiuntivi nei campi CLUSTER, CONNAME e CHANNEL per abilitare la definizione in modo che sia identica su tutti i gestori code nel cluster uniforme. Ciò comprende:

+ AUTOCL +

Il nome cluster automatico

+ NOMECODE +

Il nome del gestore code creato

+ CONNOME +

Una variabile definita durante la creazione del gestore code, utilizzando il parametro **-iv** o nella sezione `Variables qm.ini`, da utilizzare nella stringa del parametro del nome connessione. Il nome della variabile può essere qualsiasi valore.

Tenere presente che i nomi dei canali sono limitati a 20 caratteri, quindi il valore sia con gli inserimenti in, sia quando gli inserimenti vengono sostituiti, deve adattarsi a tale limitazione. Un file di esempio potrebbe essere simile al seguente:

```

*#####
* Compulsory section for all uniform cluster queue managers
*#####
define channel('+AUTOCL+_+QMNAME+') chltype(clusrcvr) trtptype(tcp) conname(+CONNAME+)
cluster('+AUTOCL+') replace
*
*#####
* Configuration for all queue managers
*#####
define QL(APPQ) maxdepth(99999999) replace
define QL(APPQ2) maxdepth(99999999) replace
define channel(CLIENTCHL) chltype(svrconn) trtptype(tcp) replace

```

3. Rendere disponibili questi due file su ciascuna macchina che ospiterà un membro cluster uniforme. Ad esempio, /shared/uniclus.ini e /shared/uniclus.mqsc.

4. Su ciascuna di queste macchine, creare il gestore code.

Sulla riga comandi, fornire:

- a. Una richiesta di avvio di un listener, sulla porta prevista
- b. Una richiesta di configurazione INI automatica (**-ii**) che punta al file di impostazione cluster automatico (uniclus.ini)
- c. Una richiesta per la configurazione MQSC automatica (**-ic**) che punta al file di configurazione MQSC che include una definizione CLUSRCVR per il cluster uniforme.
- d. Un CONNAME per questo gestore code.

Sull'host per QMA:

```

crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMA.dnsname(1414) QMA
strmqm QMA

```

Ogni gestore code nel cluster uniforme viene creato con una riga comandi quasi identica - tutte le differenze tra repository completo e parziale vengono gestite automaticamente per un cluster uniforme.

Sull'host per QMB:

```

crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMB.dnsname(1414) QMB
strmqm QMB

```

Sull'host per QMC:

```

crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMC.dnsname(1414) QMC
strmqm QMC

```

Sull'host per QMD:

```

crtmqm -p 1414 -ii /shared/uniclus.ini -ic /shared/uniclus.mqsc -iv
CONNAME=QMD.dnsname(1414) QMD
strmqm QMD

```

Cosa accade automaticamente:

Quando il gestore code viene avviato, le definizioni del file uniclus.ini vengono applicate al file qm.ini. Per ulteriori informazioni, consultare [“Configurazione automatica di qm.ini all'avvio”](#) a pagina 108. Ciò aggiunge la definizione **AutoCluster** nel file qm.ini.

Se il gestore code viene denominato nella stanza **AutoCluster** come uno dei repository completi, viene automaticamente convertito in un repository completo, simile all'emissione del comando MQSC ALTER QMGR REPOS (**ClusterName**), altrimenti viene convertito in un repository parziale, simile all'immissione del comando MQSC ALTER QMGR REPOS (").

Quando viene elaborata la definizione del canale ricevente del cluster per il cluster automatico, i canali mittenti del cluster vengono definiti da questo gestore code a tutti i repository completi nella stanza **AutoCluster** (escluso il gestore code locale se questo è uno dei repository completi). Questi canali mittenti ereditano tutti gli attributi di canale comuni dal ricevente del cluster locale che è stato definito.



Attenzione: Anche se i canali vengono creati senza ulteriore intervento manuale, questi sono oggetti del canale di gestione che possono essere visualizzati e gestiti come per qualsiasi altra definizione di canale. Non confondere questi oggetti con i canali mittente del cluster 'definiti automaticamente', creati in modo temporaneo e su richiesta dal cluster per instradare il traffico dei messaggi.

Operazioni successive

Verifica della configurazione del cluster uniforme

Quando il parametro **ClusterName** è impostato correttamente e il gestore code è un membro del cluster denominato, viene emesso il messaggio AMQ9883 per confermare che il cluster è ora identificato come un cluster uniforme.

Puoi quindi utilizzare funzionalità di cluster uniformi, come il bilanciamento automatico dell'applicazione. Durante l'avvio del gestore code, se questo parametro è stato impostato, ma il nome non è un nome cluster IBM MQ valido, il nome viene ignorato e viene emesso il messaggio di errore AMQ9882 .

Se il nome è un nome cluster valido, ma non esiste alcun canale cluster per il cluster identificato, viene emesso il messaggio di avvertenza AMQ9881 nel log degli errori del gestore code per consentire all'amministratore di identificare e correggere questa situazione.

Verifica della configurazione automatica del cluster

Se è stato utilizzato il supporto cluster automatico per impostare il cluster uniforme, è possibile verificare che i gestori code specificati come repository completi siano ora configurati correttamente, utilizzando i comandi runmqsc:

```
QMA:
  1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
  QMNAME(QMA)                                REPOS(UNICLUS)
```

Mentre i repository parziali non sono configurati come repository:

```
QMC:
  1 : dis qmgr repos
AMQ8408I: Display Queue Manager details.
  QMNAME(QMC)                                REPOS( )
```

Inoltre, dovrebbe essere possibile vedere che i canali mittente del cluster (CLUSSDR) sono stati configurati da ogni gestore code agli altri repository completi, utilizzando il nome del canale dal file MQSC di configurazione:


```
QMA:
  1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
  CHANNEL(UNICLUS_QMA)                        CHLTYPE(CLUSRCVR)
  CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
  CHANNEL(UNICLUS_QMB)                        CHLTYPE(CLUSSDR)
  CONNAME(QMB.dnsname(1414))

QMC:
  1 : dis chl(UNICLUS*) conname
AMQ8414I: Display Channel details.
  CHANNEL(UNICLUS_QMA)                        CHLTYPE(CLUSSDR)
  CONNAME(QMA.dnsname(1414))
AMQ8414I: Display Channel details.
```

CHANNEL (UNICLUS_QMB) CONNAME (QMB.dnsname(1414)) AMQ8414I: Display Channel details.	CHLTYPE (CLUSSDR)
CHANNEL (UNICLUS_QMC) CONNAME (QMC.dnsname(1414))	CHLTYPE (CLUSRCVR)

Concetti correlati

[“Informazioni sui cluster uniformi” a pagina 426](#)

L'obiettivo di una distribuzione cluster uniforme è che le applicazioni possano essere progettate per la scalabilità e la disponibilità e possano connettersi a qualsiasi gestore code all'interno del cluster uniforme. In questo modo si rimuove qualsiasi dipendenza da un gestore code specifico, migliorando la disponibilità e il bilanciamento del carico di lavoro del traffico di messaggistica.  I cluster uniformi non sono disponibili su IBM MQ for z/OS; i gruppi di condivisione code forniscono molte delle funzioni di un cluster uniforme.

[“Limitazioni e considerazioni per i cluster uniformi” a pagina 439](#)

Limitazioni e altri punti da considerare quando si configurano cluster uniformi.

Conversione di un cluster esistente in un cluster uniforme

È possibile utilizzare questa procedura per convertire un cluster esistente in un cluster uniforme.

Informazioni su questa attività

Se si converte un cluster esistente in un cluster uniforme, è necessario verificare che tutte le definizioni richieste per supportare il bilanciamento delle applicazioni tra gestori code siano presenti su tutti i membri del cluster.

Procedura

1. Abilitare la sottoscrizione di pubblicazione IBM MQ , inclusa la sottoscrizione di pubblicazione remota (in cluster) su tutti i gestori code.

Questo è un prerequisito per la funzionalità del cluster uniforme, pertanto è necessario assicurarsi che gli attributi PSMODE e PSCLUS del gestore code siano entrambi impostati sul valore predefinito ENABLED.

2. Aggiungere una sezione **AutoCluster** al file `qm.ini` al nome del cluster IBM MQ , come utilizzato nelle definizioni di oggetti MQSC, come i canali cluster.

Ad esempio, se il nome del cluster è UNICLUS, aggiungere o modificare la sezione AutoCluster nei file `qm.ini` nel modo seguente:


```
AutoCluster:  
  ClusterName=UNICLUS  
  Type=Uniform
```

3. Riavviare i gestori code per applicare la nuova impostazione.
4. Considerare l'utilizzo della configurazione automatica come meccanismo per garantire che tutti i membri del cluster uniformi abbiano la stessa configurazione applicata dall'avvio.

Consultare [Configurazione automatica da uno script MQSC all'avvio](#) per ulteriori dettagli.

Concetti correlati

[“Informazioni sui cluster uniformi” a pagina 426](#)

L'obiettivo di una distribuzione cluster uniforme è che le applicazioni possano essere progettate per la scalabilità e la disponibilità e possano connettersi a qualsiasi gestore code all'interno del cluster uniforme. In questo modo si rimuove qualsiasi dipendenza da un gestore code specifico, migliorando la disponibilità e il bilanciamento del carico di lavoro del traffico di messaggistica.  I cluster uniformi non sono disponibili su IBM MQ for z/OS; i gruppi di condivisione code forniscono molte delle funzioni di un cluster uniforme.

[“Limitazioni e considerazioni per i cluster uniformi” a pagina 439](#)

Limitazioni e altri punti da considerare quando si configurano cluster uniformi.

Multi Utilizzo della configurazione automatica del cluster

Configurare IBM MQ per abilitare la configurazione automatica modificando le informazioni di configurazione `qm.ini`.

Nota: È possibile utilizzare la stanza `AutoCluster` solo per i cluster uniformi.

Stanze da configurare

È possibile modificare le seguenti stanze:

AutoConfig

Definito nel file `qm.ini`. Quando il gestore code viene avviato, identifica quali file di configurazione automatica applicare.

Utilizzare questo meccanismo per distribuire una configurazione cluster identica quando si utilizzano cluster uniformi.

AutoCluster

Definito nel file `qm.ini`. Utilizzato quando il gestore code inizia a identificare se il cluster è un membro di un cluster automatico e può identificare i repository completi del cluster.

Variabili

Definito nel file `qm.ini`. Contiene alcune variabili del gestore code.

Attributi per la stanza `AutoConfig`

I seguenti due attributi sono consentiti nella stanza `AutoConfig`:

MQSCConfig=< Path >

Il percorso è un percorso file completo o un percorso a una directory, in cui tutti i file `*.mqsc` vengono applicati al gestore code, ad ogni avvio del gestore code.

Per ulteriori informazioni, consultare [Configurazione automatica da uno script MQSC all'avvio](#).

IniConfig=< Path >

Il percorso è un percorso file completo o un percorso a una directory, in cui tutti i file `*.ini` vengono applicati al file `qm.ini`, ad ogni avvio del gestore code.

Per ulteriori informazioni, consultare [“Configurazione automatica di qm.ini all'avvio” a pagina 108](#).

Questi attributi vengono spesso utilizzati come parte dell'impostazione di cluster uniformi. Per ulteriori informazioni, consultare [“Creazione di un nuovo cluster uniforme” a pagina 442](#).

Stanza di esempio:

```
AutoConfig:
MQSCConfig=C:\MQ_Configuration\uniclus.mqsc
IniConfig=C:\MQ_Configuration\uniclus.ini
```

Attributi per la stanza `AutoCluster`

I seguenti attributi sono obbligatori per la stanza `AutoCluster`:

Tipo =*Uniforme*

Specifica il tipo di cluster automatico e l'unica opzione valida è *Uniforme*, che rappresenta un cluster uniforme.

ClusterName=< Stringa >

Il nome del cluster, ovvero il nome del cluster automatico.

La presenza degli attributi sopra riportati abilita il bilanciamento dell'applicazione per i cluster uniformi. Consultare [“Bilanciamento automatico dell'applicazione” a pagina 428](#) per ulteriori dettagli.

Inoltre, la configurazione semplificata di un cluster può essere eseguita se il cluster è descritto in questa stanza. Per ulteriori informazioni, consultare [“Creazione di un nuovo cluster uniforme”](#) a pagina 442. Quando viene utilizzato, è possibile denominare due gestori code e fornire i relativi nomi di connessione per i repository completi per questo cluster automatico.

I seguenti attributi sono facoltativi per la stanza AutoCluster , ma è necessario fornirli in coppie:

RepositoryName1 =< Stringa>

Questo è il nome del gestore code per il primo repository completo nel cluster automatico. Può essere il nome di questo gestore code o di un altro gestore code.

Repository1Conname=< Stringa nome connessione>

Questo è il valore del nome connessione (CONNNAME) per il modo in cui i membri del cluster automatico devono connettersi a questo gestore code.

Inoltre, è possibile identificare un secondo repository completo per il cluster:

Repository2Name=< Stringa>

Repository2Conname=< Stringa nome connessione>

Stanza di esempio:

```
AutoCluster:
Repository2Conname=myFR1.hostname(1414)
Repository2Name=QMFR1
Repository1Conname= myFR2.hostname(1414)
Repository1Name=QMFR2
ClusterName=UNICLUS
Type=Uniform
```

Attributi per la stanza Variables

Una coppia attribute=value è valida nel campo attributo. Questi possono essere forniti utilizzando l'opzione della riga comandi **-iv** nel comando **crtmqm** quando si crea un gestore code.

È possibile utilizzare gli attributi elencati nella stanza Variables durante la configurazione cluster automatica di CONNAME e i campi MQSC del nome canale di un canale ricevente del cluster.

Sospensione di un gestore code da un cluster uniforme

Durante il normale funzionamento di un cluster uniforme, le istanze dell'applicazione client ricollegabili potrebbero essere automaticamente ribilanciate in qualsiasi momento, a qualsiasi gestore code nel cluster. Se si desidera impedire alle applicazioni di connettersi a un determinato gestore code per un determinato periodo di tempo, ad esempio, durante le operazioni di manutenzione o la determinazione dei problemi, utilizzare il comando **SUSPEND QMGR**.

Immettere il comando **SUSPEND QMGR CLUSTER** (*uniform cluster name*)

Oltre agli effetti tipici della sospensione da un cluster di IBM MQ , in un cluster uniforme, il comando **SUSPEND** impedisce anche che le applicazioni ricollegabili vengano ribilanciate su questo gestore code.

Qualsiasi connessione esistente al gestore code viene immediatamente ribilanciata ad altri gestori code disponibili nel cluster quando viene immesso il comando.

Note:

- Quando i gestori code vengono sospesi da un cluster, **DIS APSTATUS** li mostra come **ACTIVE (NO)**, ad eccezione del gestore code locale, che mostra sempre **ACTIVE (YES)** per la voce di stato.
- Se tutti i gestori code nel cluster uniforme sono sospesi, le applicazioni restano connesse a uno o più gestori code sospesi.

Per evitare che vengano mantenute nuove connessioni al gestore code, è necessario arrestare il canale o i canali di connessione del server utilizzati dalle proprie applicazioni client, ad esempio immettendo il seguente comando **runmqsc** :

```
STOP CHANNEL(surconn channel name)
```

Ciò potrebbe non essere possibile se, ad esempio, questi canali vengono utilizzati anche per collegare le applicazioni amministrative richieste durante la finestra di manutenzione. Per questo motivo, il gestore code sospeso controlla periodicamente le applicazioni ricollegabili collegate

Se sono presenti applicazioni ricollegabili, queste vengono ribilanciate su altri gestori code disponibili nel cluster. La manutenzione può ora essere eseguita sul gestore code sospeso.

Nota: Le applicazioni non considerate mobili non sono interessate dal comando iniziale o dalle successive scansioni e rimangono connesse al gestore code sospeso; per ulteriori dettagli, consultare [MOVCOUNT](#) .

Per riprendere un gestore code sospeso:

1. Se necessario, avviare il canale di connessione del server per riprendere ad accettare nuove connessioni dell'applicazione, immettendo il seguente comando:

```
START CHANNEL(surconn channel name)
```

2. Immettere il seguente comando **runmqsc** :

```
RESUME QMGR CLUSTER(uniform cluster name)
```

Il gestore code riprende la comunicazione con il resto del cluster uniforme e, se necessario per ripristinare il bilanciamento, le istanze dell'applicazione client riconnettibili vengono reindirizzate a questo gestore code.

Configurazione della messaggistica di pubblicazione/ sottoscrizione

È possibile avviare, arrestare e visualizzare lo stato della pubblicazione / sottoscrizione in coda. È inoltre possibile aggiungere e rimuovere i flussi e aggiungere ed eliminare i gestori code da una gerarchia broker.

Procedura

- Consultare i seguenti argomenti secondari per ulteriori informazioni sul controllo della pubblicazione / sottoscrizione accodata:
 - [“Impostazione degli attributi dei messaggi di pubblicazione / sottoscrizione accodati”](#) a pagina 448
 - [“Avvio della pubblicazione / sottoscrizione accodata”](#) a pagina 450
 - [“Arresto della pubblicazione / sottoscrizione in coda”](#) a pagina 450
 - [“Aggiunta di uno stream”](#) a pagina 451
 - [“Eliminazione di uno stream”](#) a pagina 452
 - [“Aggiunta di un punto di sottoscrizione”](#) a pagina 452
 - [“Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione”](#) a pagina 461

Impostazione degli attributi dei messaggi di pubblicazione / sottoscrizione accodati

Si controlla il funzionamento di alcuni attributi dei messaggi di pubblicazione / sottoscrizione utilizzando gli attributi del gestore code. Gli altri attributi controllati nella stanza *Broker* del file *qm.ini* .

Informazioni su questa attività

È possibile impostare i seguenti attributi di pubblicazione / sottoscrizione: per i dettagli, vedere [Parametri del gestore code](#)

Descrizione	Nome parametro MQSC
Conteggio tentativi messaggi di comando	PSRTCNT
Elimina messaggio di input di comando non consegnabile	PSNPMMSG
Comportamento che segue il messaggio di risposta del comando non distribuibile	PSNPRES
Elabora i messaggi di comando nel punto di sincronizzazione	PSSYNCPT

La stanza Broker viene utilizzata per gestire le seguenti impostazioni di configurazione:

- `PersistentPublishRetry=yes | force`

Se si specifica `Si`, se una pubblicazione di un messaggio persistente tramite l'interfaccia di pubblicazione / sottoscrizione in coda ha esito negativo e non è stata richiesta alcuna risposta negativa, l'operazione di pubblicazione viene ritentata.

Se è stato richiesto un messaggio di risposta negativa, la risposta negativa viene inviata e non si verificano ulteriori tentativi.

Se si specifica `Forza`, se una pubblicazione di un messaggio persistente tramite l'interfaccia di pubblicazione / sottoscrizione in coda non riesce, l'operazione di pubblicazione viene ritentata fino a quando non viene elaborata correttamente. Non viene inviata alcuna risposta negativa.

- `NonPersistentPublishRetry= sì | force`

Se si specifica `Si`, se una pubblicazione di un messaggio non persistente attraverso l'interfaccia di pubblicazione / sottoscrizione in coda ha esito negativo e non è stata richiesta alcuna risposta negativa, l'operazione di pubblicazione viene ritentata.

Se è stato richiesto un messaggio di risposta negativa, la risposta negativa viene inviata e non si verificano ulteriori tentativi.

Se è stato specificato `Forza`, se una pubblicazione di un messaggio non persistente tramite l'interfaccia di pubblicazione / sottoscrizione in coda ha esito negativo, l'operazione di pubblicazione viene ritentata fino a quando non viene elaborata correttamente. Non viene inviata alcuna risposta negativa.

Nota: Se si desidera abilitare questa funzionalità per i messaggi non persistenti, oltre a impostare il valore `NonPersistentPublishRetry`, è necessario assicurarsi che l'attributo del gestore code **PSSYNCPT** sia impostato su `Si`.

Questa operazione potrebbe anche avere un impatto sulle prestazioni dell'elaborazione delle pubblicazioni non persistenti poiché **MQGET** dalla coda `STREAM` ora si verifica nel punto di sincronizzazione.

- `PublishBatchDimensione =numero`

Il broker normalmente elabora i messaggi di pubblicazione all'interno del punto di sincronizzazione. Può essere inefficiente eseguire il commit di ciascuna pubblicazione singolarmente e, in alcune circostanze, il broker può elaborare più messaggi di pubblicazione in una singola unità di lavoro. Questo parametro specifica il numero massimo di messaggi di pubblicazione che possono essere elaborati in una singola unità di lavoro

Il valore predefinito per `PublishBatchDimensione` è 5.

- `PublishBatchIntervallo =numero`

Il broker normalmente elabora i messaggi di pubblicazione all'interno del punto di sincronizzazione. Può essere inefficiente eseguire il commit di ciascuna pubblicazione singolarmente e, in alcune circostanze, il broker può elaborare più messaggi di pubblicazione in una singola unità di lavoro. Questo parametro specifica il tempo massimo (in millisecondi) tra il primo messaggio in un batch e qualsiasi pubblicazione successiva inclusa nello stesso batch.

Un intervallo batch di 0 indica che è possibile elaborare fino a `PublishBatchDimensione` messaggi, purché i messaggi siano immediatamente disponibili.

Il valore predefinito per `PublishBatchIntervallo` è zero.

Procedura

Utilizzare Esplora risorse di IBM MQ , i comandi programmabili o il comando **runmqsc** per modificare gli attributi del gestore code che controllano il funzionamento della pubblicazione / sottoscrizione.

Esempio

```
PSNPRES ALTER QMGR (SAFE)
```

Avvio della pubblicazione / sottoscrizione accodata

Si avvia la pubblicazione / sottoscrizione accodata impostando l'attributo `PSMODE` del gestore code.

Prima di iniziare

Leggere la descrizione di [PSMODE](#) per comprendere le tre modalità di pubblicazione / sottoscrizione:

- COMPAT
- Disabilitato
- Abilitato

Informazioni su questa attività

Impostare l'attributo `QMGR PSMODE` per avviare l'interfaccia di pubblicazione / sottoscrizione accodata (nota anche come broker) o il motore di pubblicazione / sottoscrizione (noto anche come pubblicazione / sottoscrizione versione 7) o entrambi. Per avviare la pubblicazione / sottoscrizione accodata è necessario impostare `PSMODE` su `ENABLED`. Il valore predefinito è `ENABLED`.

Procedura

Utilizzare IBM MQ Explorer o il comando **runmqsc** per abilitare l'interfaccia di pubblicazione / sottoscrizione accodata se l'interfaccia non è già abilitata.

Esempio

```
ALTER QMGR PSMODE (ENABLED)
```

Operazioni successive

IBM MQ elabora i comandi di pubblicazione / sottoscrizione accodati e le chiamate MQI (Message Queue Interface) di pubblicazione / sottoscrizione.

Arresto della pubblicazione / sottoscrizione in coda

La pubblicazione / sottoscrizione in coda viene arrestata impostando l'attributo `PSMODE` del gestore code.

Prima di iniziare

Leggere la descrizione di [PSMODE](#) per comprendere le tre modalità di pubblicazione / sottoscrizione:

- COMPAT
- DISABILITATO
- Abilitato

Informazioni su questa attività

Impostare l'attributo PSMODE QMGR per arrestare l'interfaccia di pubblicazione / sottoscrizione accodata (nota anche come broker) o il motore di pubblicazione / sottoscrizione (noto anche come pubblicazione / sottoscrizione versione 7) o entrambi. Per arrestare la pubblicazione / sottoscrizione in coda è necessario impostare PSMODE su COMPAT. Per arrestare completamente il motore di pubblicazione / sottoscrizione, impostare PSMODE su DISABLED.

Procedura

Utilizzare IBM MQ Explorer o il comando **runmqsc** per disattivare l'interfaccia di pubblicazione / sottoscrizione in coda.

Esempio

```
ALTER QMGR PSMODE (COMPAT)
```

Aggiunta di uno stream

È possibile aggiungere i flussi manualmente per consentire l'isolamento dei dati tra le applicazioni o per consentire l'interoperatività con le gerarchie di pubblicazione / sottoscrizione IBM MQ.

Prima di iniziare

Familiarizzare con il modo in cui operano i flussi di pubblicazione / sottoscrizione. Vedi [Stream e argomenti](#).

Informazioni su questa attività

Utilizzare il comando PCF, **runmqsc** IBM MQ Explorer per eseguire queste operazioni.

Nota: È possibile eseguire i passi 1 e 2 in qualsiasi ordine. Eseguire il passo 3 solo dopo che i passi 1 e 2 sono stati entrambi completati.

Procedura

1. Definire una coda locale con lo stesso nome del flusso nella versione precedente di IBM MQ.
2. Definire un argomento locale con lo stesso nome del flusso nella versione precedente di IBM MQ.
3. Aggiungere il nome della coda all'elenco nomi, SYSTEM.QPUBSUB.QUEUE.NAMELIST
4. Ripetere per tutti i gestori code alla versione più recente di IBM MQ che si trovano nella gerarchia di pubblicazione / sottoscrizione.

Aggiunta 'Sport'

Nell'esempio di condivisione del flusso 'Sport', i gestori code della versione precedente e i gestori code della versione successiva IBM MQ operano nella stessa gerarchia di pubblicazione / sottoscrizione. I gestori code della versione precedente condividono un flusso denominato 'Sport'. L'esempio mostra come creare una coda e un argomento su gestori code di una versione più recente denominati 'Sport', con una stringa di argomenti 'Sport' condivisa con il flusso di gestori code della versione precedente 'Sport'.

Un'applicazione di pubblicazione del gestore code di una versione più recente, pubblicata nell'argomento 'Sport', con la stringa di argomenti 'Soccer/Results', crea la stringa di argomenti risultante 'Sport/Soccer/Results'. Sui gestori code della versione più recente, i sottoscrittori dell'argomento 'Sport', con stringa di argomenti 'Soccer/Results' ricevono la pubblicazione.

Sui gestori code della versione precedente, i sottoscrittori del flusso 'Sport', con la stringa di argomenti 'Soccer/Results' ricevono la pubblicazione.

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
  3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: IBM MQ namelist changed.
```

Nota: È necessario fornire sia i nomi esistenti nell'oggetto elenco nomi, sia i nuovi nomi che si stanno aggiungendo al comando **alter namelist**.

Operazioni successive

Le informazioni sul flusso vengono trasmesse ad altri broker nella gerarchia.

È necessario configurare manualmente ciascun gestore code IBM MQ nella gerarchia.

Eliminazione di uno stream

È possibile eliminare un flusso da un gestore code IBM MQ.

Prima di iniziare

Prima di eliminare un flusso, è necessario verificare che non vi siano sottoscrizioni rimanenti al flusso e disattivare tutte le applicazioni che utilizzano il flusso. Se le pubblicazioni continuano a fluire in un flusso eliminato, è necessario un notevole sforzo di gestione per ripristinare il sistema ad uno stato di funzionamento pulito.

Procedura

1. Trovare tutti i broker connessi che ospitano questo stream.
2. Annullare tutte le sottoscrizioni al flusso su tutti i broker.
3. Rimuovere la coda (con lo stesso nome del flusso) dall'elenco nomi, SYSTEM.QPUBSUB.QUEUE.NAMELIST.
4. Eliminare o eliminare tutti i messaggi dalla coda con lo stesso nome del flusso.
5. Cancellare la coda con lo stesso nome del flusso.
6. Eliminare l'oggetto argomento associato.

Operazioni successive

Ripetere i passi da 3 a 5 su tutti i gestori code IBM MQ connessi che ospitano il flusso.

Aggiunta di un punto di sottoscrizione

Come estendere un'applicazione di pubblicazione / sottoscrizione accodata esistente migrata da una versione precedente di IBM Integration Bus con un nuovo punto di sottoscrizione.

Prima di iniziare

1. Verificare che il punto di sottoscrizione non sia già definito in SYSTEM.QPUBSUB.SUBPOINT.NAMELIST.
2. Verificare se è presente un oggetto argomento o una stringa argomento con lo stesso nome del punto di sottoscrizione.


Informazioni su questa attività

IBM MQ, le applicazioni non utilizzano i punti di sottoscrizione, ma possono interagire con le applicazioni esistenti, utilizzando il meccanismo di migrazione dei punti di sottoscrizione.

Importante: Il meccanismo di migrazione del punto di sottoscrizione è stato rimosso da IBM MQ 8.0. Se è necessario migrare le applicazioni esistenti, è necessario eseguire le procedure descritte nella documentazione per la versione del prodotto, prima di migrare alla versione più recente.

Non è necessario aggiungere punti di sottoscrizione per utilizzare le applicazioni di pubblicazione / sottoscrizione integrate scritte per versioni di IBM MQ.

Procedura

1. Aggiungere il nome del punto di sottoscrizione a `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`.
 -  Su z/OS, **NLTYPE** è NONE, il valore predefinito.
 - Ripetere il passo su ogni gestore code connesso nella stessa topologia di pubblicazione / sottoscrizione.
2. Aggiungere un oggetto argomento, preferibilmente assegnandogli il nome del punto di sottoscrizione, con una stringa di argomento corrispondente al nome del punto di sottoscrizione.
 - Se il punto di sottoscrizione si trova in un cluster, aggiungere l'oggetto argomento come un argomento cluster sull'host dell'argomento cluster.
 - Se esiste un oggetto argomento con la stessa stringa argomento del nome del punto di sottoscrizione, utilizzare l'oggetto argomento esistente. È necessario comprendere le conseguenze del punto di sottoscrizione che riutilizza un argomento esistente. Se l'argomento esistente fa parte di un'applicazione esistente, è necessario risolvere il conflitto tra due argomenti con lo stesso nome.
 - Se esiste un oggetto argomento con lo stesso nome del punto di sottoscrizione, ma con una stringa argomento differente, creare un argomento con un nome diverso.
3. Impostare l'attributo **Topic** WILDCARD sul valore BLOCK.

Il blocco delle sottoscrizioni a # o * isola le sottoscrizioni con caratteri jolly ai punti di sottoscrizione, consultare [Caratteri jolly e punti di sottoscrizione](#).
4. Impostare gli attributi richiesti nell'oggetto argomento.

Esempio

L'esempio mostra un file di comandi **runmqsc** che aggiunge due punti di sottoscrizione, USD e GBP.

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

Nota:

1. Includere il punto di sottoscrizione predefinito nell'elenco di punti di sottoscrizione aggiunti utilizzando il comando **ALTER**. **ALTER** elimina i nomi esistenti nell'elenco nomi.
2. Definire gli argomenti prima di modificare l'elenco nomi. Il gestore code controlla l'elenco nomi solo quando il gestore code viene avviato e quando l'elenco nomi viene modificato.

Configurazione delle reti di pubblicazione / sottoscrizione distribuite

I gestori code connessi insieme in una topologia di pubblicazione / sottoscrizione distribuita condividono uno spazio argomenti federato comune. Le sottoscrizioni create su un gestore code possono ricevere messaggi pubblicati da un'applicazione connessa a un altro gestore code nella topologia.

È possibile controllare l'estensione degli spazi argomento creati collegando i gestori code in cluster o gerarchie. In un cluster di pubblicazione / sottoscrizione, un oggetto argomento deve essere 'in cluster'

per ogni ramo dello spazio argomento che deve estendersi al cluster. In una gerarchia, ciascun gestore code deve essere configurato per identificare il relativo 'parent' nella gerarchia.

È possibile controllare ulteriormente il flusso di pubblicazioni e sottoscrizioni all'interno della topologia, scegliendo se ciascuna pubblicazione e sottoscrizione è locale o globale. Le pubblicazioni e le sottoscrizioni locali non vengono propagate oltre il gestore code a cui è connesso il publisher o il sottoscrittore.

Concetti correlati

[Reti di pubblicazione / sottoscrizione distribuite](#)

[Ambito della pubblicazione](#)

[Ambito della sottoscrizione](#)

[Spazi argomento](#)

Attività correlate

[Definizione degli argomenti del cluster](#)

Configurazione di un cluster di pubblicazione / sottoscrizione

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER**. Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLROUTE**.

Prima di iniziare

Alcune configurazioni cluster non possono contenere i costi generali della pubblicazione / sottoscrizione instradata diretta. Prima di utilizzare questa configurazione, esplorare le considerazioni e le opzioni dettagliate in [Progettazione di cluster di pubblicazione / sottoscrizione](#).

Affinché le modifiche ad un cluster vengano propagate in tutto il cluster, deve essere sempre disponibile almeno un repository completo. Assicurarsi che i repository siano disponibili prima di avviare questa attività.

Vedere anche [Instradamento per i cluster di pubblicazione / sottoscrizione: Note sul funzionamento](#).

Scenario:

- Il cluster INVENTORY è stato configurato come descritto in “Aggiunta di un gestore code a un cluster” a pagina 329. Contiene tre gestori code; LONDON e NEWYORK contengono entrambi repository completi, PARIS contiene un repository parziale.

Informazioni su questa attività

Quando si definisce un argomento su un gestore code in un cluster, è necessario specificare se l'argomento è un argomento cluster e (in tal caso) l'instradamento all'interno del cluster per le pubblicazioni e le sottoscrizioni per questo argomento. Per rendere l'argomento un argomento cluster, configurare la proprietà **CLUSTER** sull'oggetto TOPIC con il nome del cluster. Definendo un argomento cluster su un gestore code nel cluster, l'argomento viene reso disponibile all'intero cluster. Per scegliere l'instradamento del messaggio da utilizzare nel cluster, impostare la proprietà **CLROUTE** sull'oggetto TOPIC su uno dei seguenti valori:

- **DIRECT**
- **TOPICHOST**

Per impostazione predefinita, l'instradamento argomento è **DIRECT**. Quando si configura un argomento di cluster con instradamento diretto su un gestore code, tutti i gestori code presenti nel cluster sono a conoscenza di tutti gli altri gestori code del cluster. Quando si effettuano operazioni di pubblicazione e sottoscrizione, ogni gestore code può collegarsi direttamente ad ogni altro gestore code nel cluster. Vedere [Cluster di pubblicazione / sottoscrizione instradati diretti](#).

Da IBM MQ 8.0, è invece possibile configurare l'instradamento argomento come **TOPICHOST**. Quando si utilizza l'instradamento all'host argomento, tutti i gestori code presenti nel cluster sono a conoscenza dei gestori code del cluster che ospitano le definizioni dell'argomento instradato (ossia, i gestori code

in cui è stato definito l'oggetto dell'argomento). Quando si effettuano operazioni di pubblicazione e sottoscrizione, i gestori code del cluster si connettono soltanto a questi gestori code dell'host argomento e non direttamente l'uno all'altro. I gestori code dell'host argomento sono responsabili dell'instradamento delle pubblicazioni dai gestori code su cui vengono pubblicate le pubblicazioni ai gestori code con le sottoscrizioni corrispondenti. Consultare [Cluster di pubblicazione / sottoscrizione instradati dell'host argomento](#).

Nota: Una volta che un oggetto argomento è stato raggruppato in cluster (mediante l'impostazione della proprietà **CLUSTER**) non è possibile modificare il valore della proprietà **CLROUTE**. Prima di poter modificare il valore, è necessario rimuovere l'oggetto dal cluster (**CLUSTER** impostato su ' '). La rimozione di un argomento dal cluster converte la definizione dell'argomento in un argomento locale, il che comporta un periodo durante il quale le pubblicazioni non vengono consegnate alle sottoscrizioni sui gestori code remoti; è necessario considerare questo aspetto quando si effettua questa modifica. Consultare [Effetto della definizione di un argomento non cluster con lo stesso nome di un argomento cluster di un altro gestore code](#). Se si tenta di modificare il valore della proprietà **CLROUTE** mentre è in cluster, il sistema genera un'errore MQRCCF_CLROUTE_NOT_ALTERABLE.

Procedura

1. Scegliere un gestore code per ospitare il proprio argomento.

Qualsiasi gestore code cluster può ospitare un argomento. Scegliere uno dei tre gestori code (LONDON, NEWYORK o PARIS) e configurare le proprietà dell'oggetto TOPIC. Se si prevede di utilizzare l'instradamento diretto, non fa alcuna differenza operativa quale gestore code si sceglie. Se si prevede di utilizzare l'instradamento dell'host argomento, il gestore code scelto ha ulteriori responsabilità per l'instradamento delle pubblicazioni. Pertanto, per l'instradamento dell'host dell'argomento, scegliere un gestore code che sia ospitato su uno dei sistemi più potenti e che disponga di una buona connettività di rete.

2. [Definire un argomento su un gestore code.](#)

Per rendere l'argomento un argomento cluster, includere il nome cluster quando si definisce l'argomento e impostare l'instradamento che si desidera utilizzare per pubblicazioni e sottoscrizioni per questo argomento. Ad esempio, per creare un argomento cluster di instradamento diretto sul gestore code LONDON, creare l'argomento nel modo seguente:

```
DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
```

Definendo un argomento cluster su un gestore code nel cluster, l'argomento viene reso disponibile all'intero cluster.

Per ulteriori informazioni sull'utilizzo di **CLROUTE**, consultare [DEFINE TOPIC \(CLROUTE\)](#) e [Instradamento per cluster di pubblicazione / sottoscrizione: Note sul comportamento](#).

Risultati

Il cluster è pronto a ricevere pubblicazioni e sottoscrizioni per l'argomento.

Operazioni successive

Se è stato configurato un cluster di pubblicazione / sottoscrizione instradato dell'host argomento, è probabile che si desideri aggiungere un secondo host argomento per questo argomento. Consultare [“Aggiunta di ulteriori host argomento a un cluster instradato di host argomento”](#) a pagina 458.

Se hai diversi cluster di pubblicazione / sottoscrizione separati, ad esempio perché la tua organizzazione è geograficamente dispersa, potresti voler propagare alcuni argomenti del cluster in tutti i cluster. È possibile eseguire questa operazione connettendo i cluster in una gerarchia. Consultare [“Combinazione degli spazi argomento di più cluster”](#) a pagina 463. È inoltre possibile controllare quali pubblicazioni fluiscono da un cluster all'altro. Consultare [“Combinazione e isolamento di spazi argomento in più cluster”](#) a pagina 465.

Concetti correlati

Combinazione di ambiti di pubblicazione e sottoscrizione

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

Attività correlate

Spostamento di una definizione di argomento del raggruppamento in un gestore code differente

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Aggiunta di ulteriori host argomento a un cluster instradato di host argomento

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Progettazione di cluster di pubblicazione / sottoscrizione

Risoluzione dei problemi di pubblicazione / sottoscrizione distribuiti

Blocco della pubblicazione / sottoscrizione in cluster

Spostamento di una definizione di argomento del raggruppamento in un gestore code differente

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Informazioni su questa attività

È possibile disporre di più definizioni dello stesso oggetto argomento cluster in un cluster. Si tratta di uno stato normale per un cluster instradato dell'host argomento e di uno stato insolito per un cluster instradato direttamente. Per ulteriori informazioni, consultare [Più definizioni di argomenti cluster con lo stesso nome](#).

Per spostare una definizione di argomento del cluster in un gestore code differente nel cluster senza interrompere il flusso di pubblicazioni, effettuare le operazioni riportate di seguito. La procedura sposta una definizione dal gestore code QM1 al gestore code QM2.

Procedura

1. Creare un duplicato della definizione dell'argomento cluster su QM2.

Per l'instradamento diretto, impostare tutti gli attributi in modo che corrispondano alla definizione di QM1.

Per l'instradamento dell'host argomento, definire inizialmente il nuovo host argomento come PUB (DISABLED). Ciò consente a QM2 di acquisire informazioni sulle sottoscrizioni nel cluster, ma non di avviare le pubblicazioni di instradamento.

2. Attendere la propagazione delle informazioni nel cluster.

Attendere che la nuova definizione dell'argomento cluster venga propagata dai gestori code del repository completo a tutti i gestori code nel cluster. Utilizzare il comando **DISPLAY CLUSTER** per visualizzare gli argomenti del cluster su ciascun membro del cluster e controllare una definizione originata da QM2.

Per l'instradamento dell'host argomento, attendi che il nuovo host argomento su QM2 venga a conoscenza di tutte le sottoscrizioni. Confrontare le sottoscrizioni proxy note a QM2 e quelle note a QM1. Un modo per visualizzare le sottoscrizioni proxy su un gestore code è quello di immettere il seguente comando **runmqsc** :

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

3. Per l'instradamento dell'host argomento, ridefinire l'host argomento in QM2 come PUB (ENABLED), quindi ridefinire l'host argomento in QM1 come PUB (DISABLED).

Ora che il nuovo host argomento su QM2 ha appreso di tutte le sottoscrizioni su altri gestori code, l'host argomento può avviare l'instradamento delle pubblicazioni.

Utilizzando l'impostazione PUB (DISABLED) per sospendere il traffico di messaggi tramite QM1, si garantisce che nessuna pubblicazione sia in corso tramite QM1 quando si elimina la definizione dell'argomento cluster.

4. Eliminare la definizione dell'argomento cluster da QM1.

È possibile eliminare la definizione solo da QM1 se il gestore code è disponibile. Altrimenti, è necessario eseguire entrambe le definizioni esistenti fino a quando QM1 non viene riavviato o rimosso in modo forzato.

Se QM1 rimane non disponibile per un lungo periodo di tempo e durante questo periodo di tempo è necessario modificare la definizione dell'argomento in cluster in QM2, la definizione QM2 è più recente della definizione QM1 e, di conseguenza, prevale.

Durante questo periodo, se vi sono differenze tra le definizioni su QM1 e QM2, gli errori vengono scritti nei log degli errori di entrambi i gestori code, segnalando all'utente la definizione dell'argomento cluster in conflitto.

Se QM1 non tornerà mai al cluster, ad esempio a causa di una disattivazione non prevista a seguito di un errore hardware, come ultima possibilità è possibile utilizzare il comando **RESET CLUSTER** per espellere forzatamente il gestore code. **RESET CLUSTER** elimina automaticamente tutti gli oggetti argomento ospitati sul gestore code di destinazione.

Concetti correlati

[Combinazione di ambiti di pubblicazione e sottoscrizione](#)

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

[Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione](#)

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

Attività correlate

[Configurazione di un cluster di pubblicazione / sottoscrizione](#)

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER**. Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLROUTE**.

[Aggiunta di ulteriori host argomento a un cluster instradato di host argomento](#)

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Aggiunta di ulteriori host argomento a un cluster instradato di host argomento

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

Prima di iniziare

La definizione dello stesso oggetto argomento cluster su diversi gestori code è utile solo dal punto di vista funzionale per un cluster instradato dell'host argomento. La definizione di più argomenti corrispondenti in un cluster instradato direttamente non ne modifica il funzionamento. Questa attività si applica solo ai cluster instradati dell'host argomento.

Questa attività presuppone che sia stato letto l'articolo Più definizioni di argomenti cluster con lo stesso nome, in particolare le seguenti sezioni:

- Più definizioni dell'argomento cluster in un cluster instradato all'host argomento
- Gestione speciale per il parametro PUB

Informazioni su questa attività

Quando un gestore code viene reso un host argomento instradato, deve prima conoscere l'esistenza di tutti gli argomenti correlati sottoscritti nel cluster. Se le pubblicazioni vengono pubblicate in tali argomenti nel momento in cui viene aggiunto un altro host argomento e una pubblicazione viene instradata al nuovo host prima che tale host abbia appreso dell'esistenza di sottoscrizioni su altri gestori code nel cluster, il nuovo host non inoltra tale pubblicazione a tali sottoscrizioni. Ciò fa sì che le sottoscrizioni non perda le pubblicazioni.

Le pubblicazioni non vengono instradate tramite gestore code dell'host argomento che hanno impostato esplicitamente il parametro **PUB** dell'oggetto argomento cluster su **DISABLED**, pertanto è possibile utilizzare questa impostazione per garantire che nessuna sottoscrizione perda le pubblicazioni durante il processo di aggiungere un ulteriore host argomento.

Nota: Mentre un gestore code ospita un argomento del cluster definito come **PUB (DISABLED)**, i publisher connessi a tale gestore code non possono pubblicare i messaggi e le sottoscrizioni corrispondenti su tale gestore code non ricevono le pubblicazioni pubblicate su altri gestori code del cluster. Per questo motivo, è necessario considerare attentamente la definizione degli argomenti instradati dell'host argomento sui gestori code in cui esistono le sottoscrizioni e la connessione delle applicazioni di pubblicazione.

Procedura

1. Configurare un nuovo host argomento e definire inizialmente il nuovo host argomento come PUB(DISABLED).

Ciò consente al nuovo host argomento di conoscere le sottoscrizioni nel cluster, ma non di avviare le pubblicazioni di instradamento.

Per informazioni sulla configurazione di un host argomento, consultare [“Configurazione di un cluster di pubblicazione / sottoscrizione”](#) a pagina 454.

2. Determinare quando il nuovo host argomento ha appreso tutte le sottoscrizioni.

A tale scopo, confrontare le sottoscrizioni proxy note al nuovo host argomento e quelle note all'host argomento esistente. Un modo per visualizzare le sottoscrizioni proxy è immettere il seguente comando **runmqsc** : DISPLAY SUB(*) SUBTYPE(PROXY)

3. Ridefinire il nuovo host argomento come PUB(ENABLED).

Dopo che il nuovo host argomento ha appreso di tutte le sottoscrizioni su altri gestori code, l'argomento può avviare l'instradamento delle pubblicazioni.

Concetti correlati

Combinazione di ambiti di pubblicazione e sottoscrizione

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

Attività correlate

Configurazione di un cluster di pubblicazione / sottoscrizione

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER** . Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLRROUTE** .

Spostamento di una definizione di argomento del raggruppamento in un gestore code differente

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Combinazione di ambiti di pubblicazione e sottoscrizione

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

Le pubblicazioni possono essere destinate a tutti i gestori code connessi in una topologia di pubblicazione / sottoscrizione o solo al gestore code locale. Allo stesso modo per le sottoscrizioni proxy. Le pubblicazioni corrispondenti a una sottoscrizione sono regolate dalla combinazione di questi due flussi.

Le pubblicazioni e le sottoscrizioni possono essere entrambe nell'ambito di QMGR o ALL. Se un publisher e un sottoscrittore sono entrambi connessi allo stesso gestore code, le impostazioni dell'ambito non influiscono sulle pubblicazioni che il sottoscrittore riceve da tale publisher.

Se il publisher e il sottoscrittore sono connessi a gestori code differenti, entrambe le impostazioni devono essere ALL per ricevere le pubblicazioni remote.

Si supponga che i publisher siano connessi a gestori code differenti. Se si desidera che un sottoscrittore riceva le pubblicazioni da un qualsiasi publisher, impostare l'ambito della sottoscrizione su ALL. È quindi possibile decidere, per ciascun publisher, se limitare l'ambito delle proprie pubblicazioni ai sottoscrittori locali per il publisher.

Si supponga che i sottoscrittori siano connessi a gestori code differenti. Se si desidera che le pubblicazioni da un publisher vengano inviate a tutti i sottoscrittori, impostare l'ambito della pubblicazione su ALL. Se si desidera che un sottoscrittore riceva le pubblicazioni solo da un publisher connesso allo stesso gestore code, impostare l'ambito della sottoscrizione su QMGR.

Esempio: servizio risultati di calcio

Supponiamo che tu sia un membro di una squadra di calcio. Ogni team ha un gestore code connesso a tutti gli altri team in un cluster di pubblicazione / sottoscrizione.

Le squadre pubblicano i risultati di tutte le partite giocate in casa utilizzando l'argomento, `Football/result/Home team name/Away team name`. Le stringhe in corsivo sono nomi argomento variabili e la pubblicazione è il risultato della corrispondenza.

Ogni club ripubblica i risultati solo per il club utilizzando la stringa di argomento `Football/myteam/Home team name/Away team name`.

Entrambi gli argomenti vengono pubblicati nell'intero cluster.

Le seguenti sottoscrizioni sono state stabilite dalla lega in modo che i tifosi di qualsiasi squadra possano iscriversi ai risultati in tre modi interessanti.

Nota che puoi configurare gli argomenti del cluster con SUBSCOPE (QMGR). Le definizioni di argomento vengono propagate a ciascun membro del cluster, ma l'ambito della sottoscrizione è solo il gestore code locale. Pertanto, i sottoscrittori a ciascun gestore code ricevono pubblicazioni diverse dalla stessa sottoscrizione.

Ricevi tutti i risultati

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

Ricevi tutti i risultati a casa

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

Poiché la sottoscrizione ha un ambito QMGR, vengono messi in corrispondenza solo i risultati pubblicati a terra.

Ricevi tutti i risultati dei miei team

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

Poiché la sottoscrizione ha un ambito QMGR, vengono messi in corrispondenza solo i risultati del team locale, ripubblicati localmente.

Concetti correlati

Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

Reti di pubblicazione / sottoscrizione distribuite

Ambito della pubblicazione

Ambito della sottoscrizione

Attività correlate

Configurazione di un cluster di pubblicazione / sottoscrizione

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER** . Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLROUTE** .

Spostamento di una definizione di argomento del raggruppamento in un gestore code differente

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Aggiunta di ulteriori host argomento a un cluster instradato di host argomento

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

È possibile creare diversi spazi argomenti di pubblicazione / sottoscrizione utilizzando i blocchi di creazione degli attributi **CLUSTER**, **PUBSCOPE** e **SUBSCOPE** , i cluster di pubblicazione / sottoscrizione e le gerarchie di pubblicazione / sottoscrizione.

Partendo dall'esempio del ridimensionamento incrementale da un singolo gestore code a un cluster di pubblicazione / sottoscrizione, i seguenti scenari illustrano diverse topologie di pubblicazione / sottoscrizione.

Concetti correlati

Combinazione di ambiti di pubblicazione e sottoscrizione

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

Reti di pubblicazione / sottoscrizione distribuite

Spazi argomento

Attività correlate

Configurazione di un cluster di pubblicazione / sottoscrizione

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER** . Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLROUTE** .

Spostamento di una definizione di argomento del raggruppamento in un gestore code differente

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Aggiunta di ulteriori host argomento a un cluster instradato di host argomento

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in

cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Definizione degli argomenti del cluster

Creazione di un singolo spazio argomento in un cluster di pubblicazione / sottoscrizione

Ridimensionare un sistema di pubblicazione / sottoscrizione per l'esecuzione su più gestori code. Utilizzare un cluster di pubblicazione / sottoscrizione per fornire a ciascun publisher e sottoscrittore un singolo spazio argomenti identico.

Prima di iniziare

È stato implementato un sistema di pubblicazione / sottoscrizione su un unico gestore code versione 7.

Creare sempre spazi argomento con i propri argomenti root, piuttosto che affidarsi all'eredità degli attributi di SYSTEM.BASE.TOPIC. Se si ridimensiona il sistema di pubblicazione / sottoscrizione fino a un cluster, è possibile definire i propri argomenti root come argomenti cluster, sull'host dell'argomento cluster e quindi tutti gli argomenti vengono condivisi all'interno del cluster.

Informazioni su questa attività

Ora si desidera scalare il sistema per supportare più publisher e sottoscrittori e avere ogni argomento visibile in tutto il cluster.

Procedura

1. Creare un cluster da utilizzare con il sistema di pubblicazione / sottoscrizione.
Se si dispone di un cluster tradizionale esistente, per motivi di prestazioni è preferibile impostare un nuovo cluster per il nuovo sistema di pubblicazione - sottoscrizione. È possibile utilizzare gli stessi server per i repository cluster di entrambi i cluster
2. Scegliere un gestore code, possibilmente uno dei repository, come host dell'argomento del cluster.
3. Assicurarsi che ogni argomento che deve essere visibile in tutto il cluster di pubblicazione / sottoscrizione si risolva in un oggetto argomento di gestione.
Impostare l'attributo **CLUSTER** denominando il cluster di pubblicazione / sottoscrizione.

Operazioni successive

Connettere le applicazioni publisher e sottoscrittore a qualsiasi gestore code nel cluster.

Creare oggetti argomento di gestione con l'attributo **CLUSTER**. Gli argomenti vengono anche propagati in tutto il cluster. I programmi di pubblicazione e sottoscrizione utilizzano gli argomenti di gestione in modo che il loro comportamento non venga modificato dalla connessione a gestori code differenti nel cluster

Se è necessario che SYSTEM.BASE.TOPIC agisca come un argomento del cluster su ogni gestore code, è necessario modificarlo su ogni gestore code.

Concetti correlati

Reti di pubblicazione / sottoscrizione distribuite

Spazi argomento

Attività correlate

Combinazione degli spazi argomento di più cluster

Creare spazi argomento che si estendono su più cluster. Pubblicare in un argomento in un cluster e sottoscriverlo in un'altro.

Combinazione e isolamento di spazi argomento in più cluster

Isolare alcuni spazi argomento in un cluster specifico e combinare altri spazi argomento per renderli accessibili in tutti i cluster connessi.

Pubblicazione e sottoscrizione di spazi argomento in più cluster

Pubblicare e sottoscrivere argomenti in più cluster utilizzando cluster sovrapposti. È possibile utilizzare questa tecnica purché gli spazi argomento nei cluster non si sovrappongano.

Definizione degli argomenti del cluster

Combinazione degli spazi argomento di più cluster

Creare spazi argomento che si estendono su più cluster. Pubblicare in un argomento in un cluster e sottoscriverlo in un'altro.

Prima di iniziare

Questa attività presuppone che si disponga di cluster di pubblicazione / sottoscrizione instradati direttamente e che si desidera propagare alcuni argomenti del cluster in tutti i cluster.

Nota: Non è possibile eseguire questa operazione per i cluster di pubblicazione / sottoscrizione instradati dell'host argomento.

Informazioni su questa attività

Per propagare le pubblicazioni da un cluster a un altro, è necessario unire i cluster in una gerarchia; consultare [Figura 65 a pagina 464](#). Le connessioni gerarchiche propagano le sottoscrizioni e pubblicazioni tra i gestori code connessi e i cluster propagano gli argomenti del cluster all'interno di ogni cluster, ma non tra i cluster.

La combinazione di questi due meccanismi propaga gli argomenti del cluster tra tutti i cluster. È necessario ripetere le definizioni degli argomenti del cluster in ogni cluster.

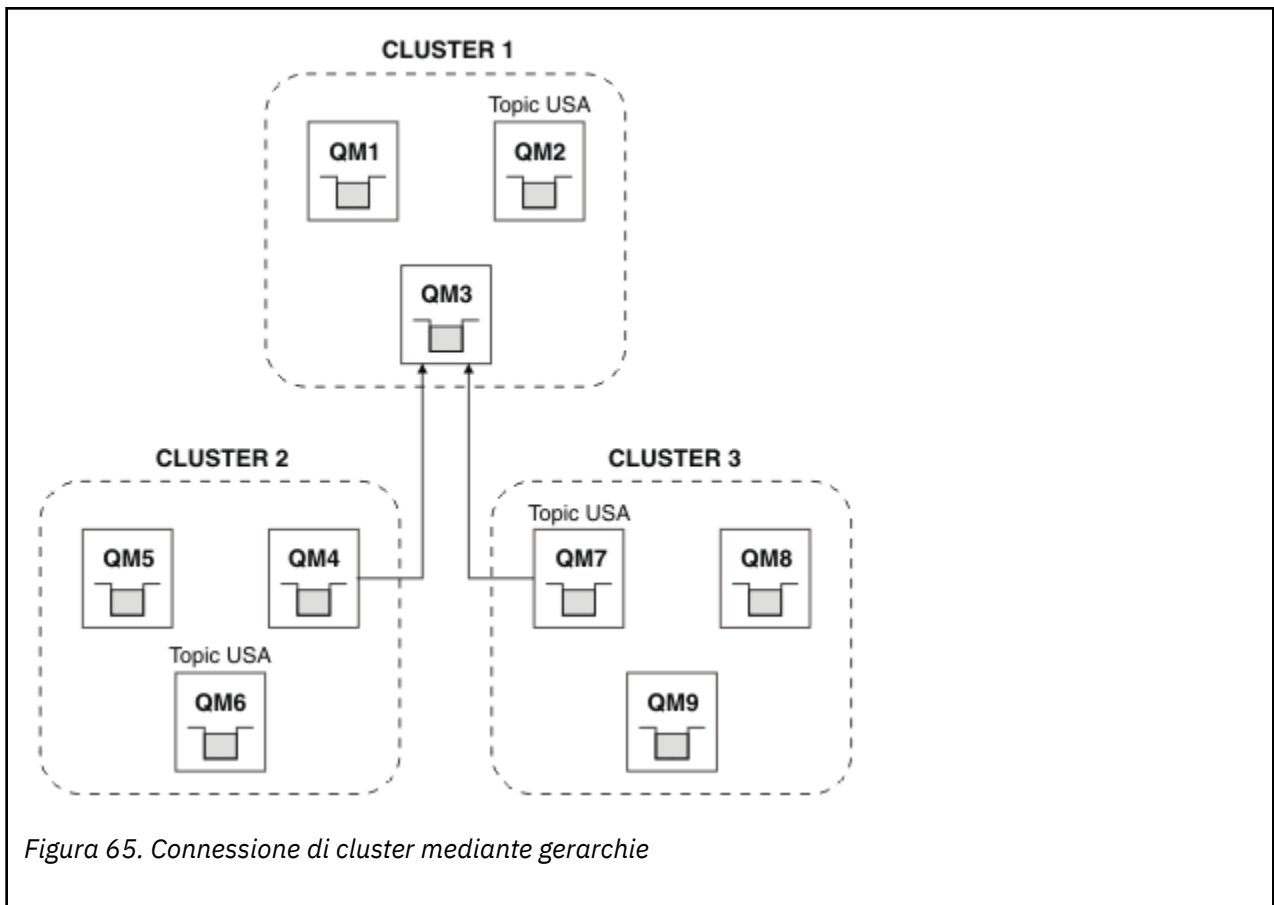


Figura 65. Connessione di cluster mediante gerarchie

La seguente procedura collega i cluster in una gerarchia.

Procedura

1. Creare due serie di canali mittente - ricevente per collegare QM3 e QM4, e QM3 e QM7, in entrambe le direzioni. È necessario utilizzare i canali mittente - destinatario tradizionali e le code di trasmissione, piuttosto che un cluster, per collegare una gerarchia.
2. Creare tre code di trasmissione con i nomi dei gestori code di destinazione. Utilizzare gli alias del gestore code se non è possibile utilizzare il nome del gestore code di destinazione come nome della coda di trasmissione per qualche motivo.
3. Configurare le code di trasmissione per attivare i canali mittente.
4. Verificare che **PSMODE** di QM3, QM4 e QM7 sia impostata su ENABLE.
5. Modificare l'attributo **PARENT** di QM4 e QM7 in QM3.
6. Verificare che lo stato della relazione padre - figlio tra i gestori code sia attivo in entrambe le direzioni.
7. Creare l'argomento di gestione USA con l'attributo **CLUSTER** (' CLUSTER 1 '), **CLUSTER** (' CLUSTER 2 ') e **CLUSTER** (' CLUSTER 3 ') su ognuno dei tre gestori code dell'argomento del cluster nei cluster 1, 2 e 3. L'host argomento del cluster non deve essere un gestore code connesso gerarchicamente.

Operazioni successive

È ora possibile pubblicare o sottoscrivere l'argomento del cluster USA in [Figura 65 a pagina 464](#). Le sottoscrizioni delle pubblicazioni vengono inviate ai publisher e ai sottoscrittori in tutti e tre i cluster.

Si supponga di non aver creato USA come argomento cluster negli altri cluster. Se USA è definito solo su QM7, le pubblicazioni e le sottoscrizioni a USA vengono scambiate tra QM7, QM8, QM9 e QM3. I publisher e i sottoscrittori in esecuzione su QM7, QM8, QM9 ereditano gli attributi dell'argomento di gestione USA. I publisher e i sottoscrittori su QM3 ereditano gli attributi di SYSTEM.BASE.TOPIC su QM3.

Vedi anche ["Combinazione e isolamento di spazi argomento in più cluster"](#) a pagina 465.

Concetti correlati

[Reti di pubblicazione / sottoscrizione distribuite](#)

[Spazi argomento](#)

Attività correlate

[Creazione di un singolo spazio argomento in un cluster di pubblicazione / sottoscrizione](#)

Ridimensionare un sistema di pubblicazione / sottoscrizione per l'esecuzione su più gestori code.

Utilizzare un cluster di pubblicazione / sottoscrizione per fornire a ciascun publisher e sottoscrittore un singolo spazio argomenti identico.

[Combinazione e isolamento di spazi argomento in più cluster](#)

Isolare alcuni spazi argomento in un cluster specifico e combinare altri spazi argomento per renderli accessibili in tutti i cluster connessi.

[Pubblicazione e sottoscrizione di spazi argomento in più cluster](#)

Pubblicare e sottoscrivere argomenti in più cluster utilizzando cluster sovrapposti. È possibile utilizzare questa tecnica purché gli spazi argomento nei cluster non si sovrappongano.

[Definizione degli argomenti del cluster](#)

Combinazione e isolamento di spazi argomento in più cluster

Isolare alcuni spazi argomento in un cluster specifico e combinare altri spazi argomento per renderli accessibili in tutti i cluster connessi.

Prima di iniziare

Esaminare l'argomento "[Combinazione degli spazi argomento di più cluster](#)" a pagina 463. Potrebbe essere sufficiente per le tue esigenze, senza aggiungere un ulteriore gestore code come bridge.

Nota: È possibile completare questa attività solo utilizzando cluster di pubblicazione / sottoscrizione instradati diretti. Non è possibile eseguire questa operazione utilizzando i cluster instradati dell'host argomento.

Informazioni su questa attività

Un potenziale miglioramento nella topologia mostrata in [Figura 65 a pagina 464](#) in "[Combinazione degli spazi argomento di più cluster](#)" a pagina 463 è quello di isolare gli argomenti del cluster che non sono condivisi tra tutti i cluster. Isolare i cluster creando un gestore code di collegamento che non si trova in nessuno dei cluster; consultare [Figura 66 a pagina 466](#). Utilizzare il gestore code di collegamento per filtrare le pubblicazioni e le sottoscrizioni che possono passare da un cluster all'altro.

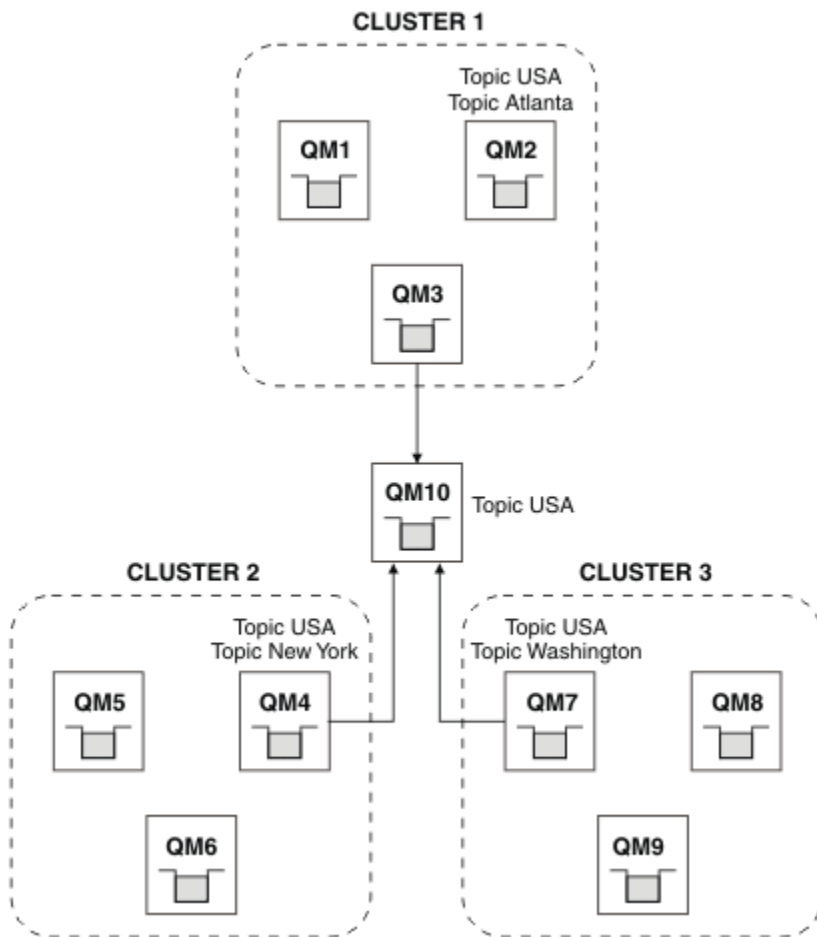


Figura 66. Cluster con bridge

Utilizzare il bridge per isolare gli argomenti del cluster che non si desidera vengano esposti attraverso il bridge sugli altri cluster. In Figura 66 a pagina 466, USA è un argomento cluster condiviso in tutti i cluster e Atlanta, New York e Washington sono argomenti cluster condivisi solo in un cluster ciascuno.

Modellare la propria configurazione utilizzando la seguente procedura:

Procedura

1. Modificare tutti gli oggetti dell'argomento SYSTEM.BASE.TOPIC in **SUBSCOPE** (QMGR) e **PUBSCOPE** (QMGR) su tutti i gestori code.
Nessun argomento (anche argomenti cluster) viene propagato ad altri gestori code a meno che non si imposti esplicitamente **SUBSCOPE** (ALL) e **PUBSCOPE** (ALL) sull'argomento root degli argomenti del tuo cluster.
2. Definire gli argomenti sui tre gestori code dell'host del cluster che si desidera condividere in ciascun cluster con gli attributi **CLUSTER** (clustername), **SUBSCOPE** (ALL) e **PUBSCOPE** (ALL).
Se si desidera che alcuni argomenti del cluster vengano condivisi tra tutti i cluster, definire lo stesso argomento in ciascuno dei cluster. Utilizzare il nome cluster di ciascun cluster come attributo cluster.
3. Per gli argomenti cluster che si desidera condividere tra tutti i cluster, definire di nuovo gli argomenti sul gestore code bridge (QM10), con gli attributi **SUBSCOPE** (ALL) e **PUBSCOPE** (ALL).

Esempio

Nell'esempio in Figura 66 a pagina 466, solo gli argomenti che ereditano da USA si propagano tra tutti i tre cluster.

Operazioni successive

Sottoscrizioni per argomenti definiti sul gestore code del bridge con **SUBSCOPE** (ALL) e **PUBSCOPE** (ALL) sono propagati tra i cluster.

Sottoscrizioni per argomenti definiti all'interno di ciascun cluster con attributi **CLUSTER** (*clustername*), **SUBSCOPE** (ALL) e **PUBSCOPE** (ALL) vengono propagati all'interno di ciascun cluster.

Le altre sottoscrizioni sono locali per un gestore code.

Concetti correlati

[Reti di pubblicazione / sottoscrizione distribuite](#)

[Spazi argomento](#)

[Ambito della pubblicazione](#)

[Ambito della sottoscrizione](#)

Attività correlate

[Creazione di un singolo spazio argomento in un cluster di pubblicazione / sottoscrizione](#)

[Ridimensionare un sistema di pubblicazione / sottoscrizione per l'esecuzione su più gestori code.](#)

[Utilizzare un cluster di pubblicazione / sottoscrizione per fornire a ciascun publisher e sottoscrittore un singolo spazio argomenti identico.](#)

[Combinazione degli spazi argomento di più cluster](#)

[Creare spazi argomento che si estendono su più cluster. Pubblicare in un argomento in un cluster e sottoscriverlo in un'altro.](#)

[Pubblicazione e sottoscrizione di spazi argomento in più cluster](#)

[Pubblicare e sottoscrivere argomenti in più cluster utilizzando cluster sovrapposti. È possibile utilizzare questa tecnica purché gli spazi argomento nei cluster non si sovrappongano.](#)

[Definizione degli argomenti del cluster](#)

Pubblicazione e sottoscrizione di spazi argomento in più cluster

[Pubblicare e sottoscrivere argomenti in più cluster utilizzando cluster sovrapposti. È possibile utilizzare questa tecnica purché gli spazi argomento nei cluster non si sovrappongano.](#)

Prima di iniziare

Creare più cluster tradizionali con alcuni gestori code nelle intersezioni tra i cluster.

Informazioni su questa attività

È possibile che si sia scelto di sovrapporre i cluster per diversi motivi.

1. Si dispone di un numero limitato di server ad alta disponibilità o di gestori code. Si decide di distribuire tutti i repository del cluster e gli host degli argomenti del cluster.
2. Si dispone di cluster di gestori code tradizionali esistenti connessi mediante gestori code gateway. Si desidera distribuire le applicazioni di pubblicazione / sottoscrizione alla stessa topologia cluster.
3. Si dispone di diverse applicazioni di pubblicazione / sottoscrizione autonome. Per motivi di prestazioni, è preferibile mantenere i cluster di pubblicazione / sottoscrizione piccoli e separati dai cluster tradizionali. Si è deciso di distribuire le applicazioni a cluster differenti. Tuttavia, si desidera monitorare anche tutte le applicazioni di pubblicazione / sottoscrizione su un gestore code, poiché si dispone di una sola copia dell'applicazione di controllo. Questo gestore code deve avere accesso alle pubblicazioni per raggruppare gli argomenti in tutti i cluster.

Assicurando che gli argomenti siano definiti in spazi argomento non sovrapposti, è possibile distribuire gli argomenti in cluster di pubblicazione / sottoscrizione sovrapposti, consultare Figura 67 a pagina 468. Se gli spazi argomento si sovrappongono, la distribuzione ai cluster che si sovrappongono causa problemi.

Poiché i cluster di pubblicazione / sottoscrizione si sovrappongono, è possibile pubblicare e sottoscrivere uno qualsiasi degli spazi argomento utilizzando i gestori code nella sovrapposizione.

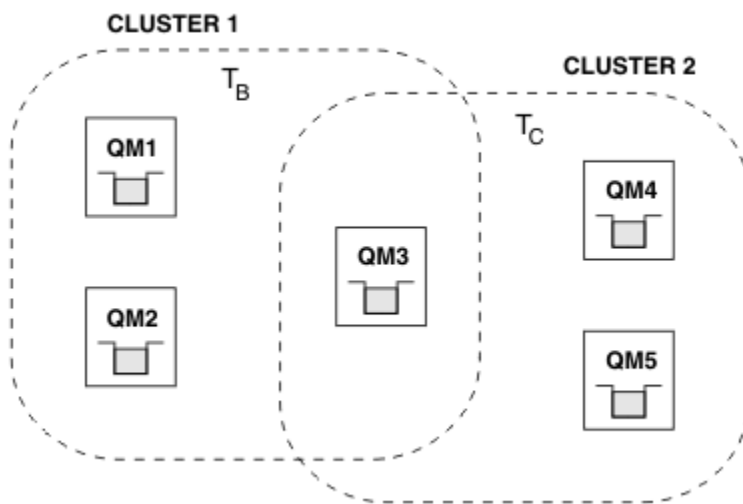


Figura 67. Cluster sovrapposti, spazi argomento non sovrapposti

Procedura

Creare un mezzo per garantire che gli spazi argomento non si sovrappongano.

Ad esempio, definire un argomento root univoco per ciascuno degli spazi argomento. Creare gli argomenti del cluster degli argomenti root.

- a) DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...
- b) DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...

Esempio

In [Figura 67 a pagina 468](#) i publisher e i sottoscrittori collegati a QM3 possono pubblicare o sottoscrivere T_B o T_C

Operazioni successive

Connettere i publisher e i sottoscrittori che utilizzano argomenti in entrambi i cluster ai gestori code nella sovrapposizione.

Connettere i publisher e i sottoscrittori che devono utilizzare solo argomenti in un cluster specifico ai gestori code che non si sovrappongono.

Concetti correlati

[Reti di pubblicazione / sottoscrizione distribuite](#)

[Spazi argomento](#)

Attività correlate

[Creazione di un singolo spazio argomento in un cluster di pubblicazione / sottoscrizione](#)

[Ridimensionare un sistema di pubblicazione / sottoscrizione per l'esecuzione su più gestori code.](#)

[Utilizzare un cluster di pubblicazione / sottoscrizione per fornire a ciascun publisher e sottoscrittore un singolo spazio argomenti identico.](#)

[Combinazione degli spazi argomento di più cluster](#)

Creare spazi argomento che si estendono su più cluster. Pubblicare in un argomento in un cluster e sottoscriverlo in un'altro.

[Combinazione e isolamento di spazi argomento in più cluster](#)

Isolare alcuni spazi argomento in un cluster specifico e combinare altri spazi argomento per renderli accessibili in tutti i cluster connessi.

[Definizione degli argomenti del cluster](#)

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

Prima di iniziare

1. I gestori code in una gerarchia di pubblicazione / sottoscrizione devono avere nomi gestore code univoci.
2. Una gerarchia di pubblicazione / sottoscrizione si basa sulla funzionalità del gestore code "pubblicazione / sottoscrizione accodata". Deve essere abilitato sia sul gestore code principale che su quello secondario. Consultare ["Avvio della pubblicazione / sottoscrizione accodata"](#) a pagina 450.
3. La relazione di pubblicazione / sottoscrizione si basa sui canali mittente e destinatario del gestore code. Ci sono due modi per stabilire i canali:
 - Aggiungere i gestori code parent e child a un cluster IBM MQ. Consultare ["Aggiunta di un gestore code a un cluster"](#) a pagina 329.
 - Stabilire una coppia di canali mittente / ricevente dal gestore code child al parent e dal parent al child. Ciascun canale deve utilizzare una coda di trasmissione con lo stesso nome del gestore code di destinazione o un alias del gestore code con lo stesso nome del gestore code di destinazione. Per ulteriori informazioni su come stabilire una connessione del canale point-to-point, consultare ["Tecniche di accodamento distribuito IBM MQ"](#) a pagina 206.

Per esempi che configurano una gerarchia su ciascun tipo di configurazione del canale, consultare la seguente serie di scenari della gerarchia di pubblicazione / sottoscrizione:

- [Scenario 1: utilizzo di canali point - to - point con alias del nome gestore code](#)
- [Scenario 2: utilizzo di canali point-to-point con lo stesso nome per la coda di trasmissione e il gestore code remoto](#)
- [Scenario 3: utilizzo di un canale cluster per aggiungere un gestore code](#)

Informazioni su questa attività

Utilizzare il comando ALTER QMGR PARENT (*PARENT_NAME*) **runmqsc** per collegare i figli ai padri. Questa configurazione viene eseguita sul gestore code child, dove *PARENT_NAME* è il nome del gestore code parent.

Procedura

```
ALTER QMGR PARENT (PARENT_NAME)
```

Esempio

Il primo esempio mostra come collegare il gestore code QM2 come elemento secondario di QM1, quindi interrogare QM2 per confermare che è diventato un elemento secondario con **STATUS ACTIVE**:

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: IBM MQ queue manager changed.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
```

QMNAME(QM1) STATUS(ACTIVE)	TYPE(PARENT)
-------------------------------	--------------

L'esempio successivo mostra il risultato della query di QM1 per le relative connessioni:

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
  QMNAME(QM1) TYPE(LOCAL)
  STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
  QMNAME(QM2) TYPE(CHILD)
  STATUS(ACTIVE)
```

Se **STATUS** non viene visualizzato come ATTIVO, verificare che i canali tra l'elemento child e l'elemento parent siano correttamente configurati e in esecuzione. Controllare entrambi i log degli errori del gestore code per eventuali errori.

Operazioni successive

Per impostazione predefinita, gli argomenti utilizzati dai publisher e dai sottoscrittori su un gestore code vengono condivisi con i publisher e con i sottoscrittori sugli altri gestori code nella gerarchia. Gli argomenti gestiti possono essere configurati per controllare il livello di condivisione tramite l'utilizzo delle proprietà degli argomenti **SUBSCOPE** e **PUBSCOPE**. Consultare [“Configurazione delle reti di pubblicazione / sottoscrizione distribuite”](#) a pagina 453.

Concetti correlati

[Combinazione di ambiti di pubblicazione e sottoscrizione](#)

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

[Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione](#)

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

Attività correlate

[Configurazione di un cluster di pubblicazione / sottoscrizione](#)

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER**. Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLROUTE**.

[Spostamento di una definizione di argomento del raggruppamento in un gestore code differente](#)

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

[Aggiunta di ulteriori host argomento a un cluster instradato di host argomento](#)

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

[Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione](#)

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Riferimenti correlati

[Stream e argomenti](#)

[VISUALIZZA PUBSUB](#)

Disconnessione di un gestore code da una gerarchia di pubblicazione / sottoscrizione

Disconnettere un gestore code secondario da un gestore code principale in una gerarchia di pubblicazione / sottoscrizione.

Informazioni su questa attività

Utilizzare il comando **ALTER QMGR** per disconnettere un gestore code da una gerarchia broker. È possibile disconnettere un gestore code in qualsiasi ordine in qualsiasi momento.

La richiesta corrispondente di aggiornare l'elemento principale viene inviata quando la connessione tra i gestori code è in esecuzione.

Procedura

```
ALTER QMGR PARENT( '')
```

Esempio

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent('')
AMQ8005: IBM MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: IBM MQ object not found.
display pubsub type(parent)
  3 : display pubsub type(parent)
AMQ8147: IBM MQ object not found.
```

Operazioni successive

È possibile eliminare tutti i flussi, le code e i canali definiti manualmente che non sono più necessari.

Concetti correlati

Combinazione di ambiti di pubblicazione e sottoscrizione

Da IBM WebSphere MQ 7.0 in poi, la pubblicazione e l'ambito della sottoscrizione funzionano in modo indipendente per determinare il flusso di pubblicazioni tra i gestori code.

Combinazione di spazi argomento nelle reti di pubblicazione / sottoscrizione

Combinare lo spazio argomenti di un gestore code con altri gestori code in una gerarchia o in un cluster di pubblicazione / sottoscrizione. Combinare cluster di pubblicazione / sottoscrizione e cluster di pubblicazione / sottoscrizione con le gerarchie.

Attività correlate

Configurazione di un cluster di pubblicazione / sottoscrizione

Definire un argomento su un gestore code. Per rendere l'argomento un argomento cluster, impostare la proprietà **CLUSTER** . Per scegliere l'instradamento da utilizzare per pubblicazioni e sottoscrizioni per questo argomento, impostare la proprietà **CLRROUTE** .

Spostamento di una definizione di argomento del raggruppamento in un gestore code differente

Per l'host argomento instradato o i cluster instradati direttamente, potrebbe essere necessario spostare una definizione dell'argomento del cluster durante la disattivazione di un gestore code o perché un gestore code del cluster ha riportato un errore o non è disponibile per un periodo di tempo significativo.

Aggiunta di ulteriori host argomento a un cluster instradato di host argomento

In un cluster di pubblicazione / sottoscrizione instradato dall'host argomento, è possibile utilizzare più gestori code per instradare le pubblicazioni alle sottoscrizioni definendo lo stesso oggetto argomento in cluster su tali gestori code. Può essere utilizzato per migliorare la disponibilità e il bilanciamento del carico di lavoro. Quando si aggiunge un host argomento supplementare per lo stesso oggetto argomento

cluster, è possibile utilizzare il parametro **PUB** per controllare quando le pubblicazioni iniziano ad essere instradate attraverso il nuovo host argomento.

Connessione di un gestore code a una gerarchia di pubblicazione / sottoscrizione

Si connette il gestore code secondario al gestore code principale nella gerarchia. Se il gestore code child è già un membro di un'altra gerarchia o cluster, questa connessione unisce le gerarchie o unisce il cluster alla gerarchia.

ALW Configurazione di più installazioni

Quando si utilizzano più installazioni sullo stesso sistema, è necessario configurare le installazioni e i gestori code.

Informazioni su questa attività

Queste informazioni si applicano a AIX, Linux, and Windows.

Procedura

- Utilizzare le informazioni contenute nei seguenti collegamenti per configurare le installazioni:
 - [“Modifica dell'installazione primaria” a pagina 479](#)
 - [“Associazione di un gestore code a un'installazione” a pagina 481](#)
 - [“Connessione di applicazioni in un ambiente di installazione multiplo” a pagina 472](#)

ALW Connessione di applicazioni in un ambiente di installazione multiplo

Sui sistemi AIX, Linux, and Windows , se le librerie IBM MQ vengono caricate, IBM MQ utilizza automaticamente le librerie appropriate senza che sia necessario eseguire ulteriori operazioni. IBM MQ utilizza le librerie dell'installazione associate con il gestore code a cui si connette l'applicazione.

I seguenti concetti vengono utilizzati per spiegare il modo in cui le applicazioni si collegano a IBM MQ:

collegamento

Quando l'applicazione viene compilata, l'applicazione viene collegata alle librerie IBM MQ per ottenere le esportazioni di funzioni che vengono caricate quando l'applicazione viene eseguita.

Caricamento

Quando l'applicazione viene eseguita, le librerie IBM MQ vengono ubicate e caricate. Il meccanismo specifico utilizzato per individuare le librerie varia in base al sistema operativo e al modo in cui viene creata l'applicazione. Per ulteriori informazioni su come individuare e caricare le librerie in un ambiente di installazione multipla, consultare [“Caricamento delle librerie IBM MQ” a pagina 473](#).

In fase di connessione

Quando l'applicazione si connette a un gestore code in esecuzione, ad esempio, utilizzando una chiamata MQCONN o MQCONNX , si connette utilizzando le librerie IBM MQ caricate.

Quando un'applicazione server si connette a un gestore code, le librerie caricate devono provenire dall'installazione associata al gestore code. Con più installazioni su un sistema, questa restrizione introduce nuove domande di verifica quando si sceglie il meccanismo utilizzato dal sistema operativo per individuare le librerie IBM MQ da caricare:

- Quando il comando **setmqm** viene utilizzato per cambiare l'installazione associata a un gestore code, le librerie che devono essere caricate cambiano.
- Quando un'applicazione si connette a più gestori code appartenenti a installazioni differenti, è necessario caricare più serie di librerie.

Tuttavia, se IBM MQ, librerie, sono ubicate e caricate, IBM MQ carica e utilizza le librerie appropriate senza che sia necessario intraprendere ulteriori azioni. Quando l'applicazione si connette a un gestore code IBM MQ carica le librerie dall'installazione a cui è associato il gestore code.

Gli scenari di migrazione e la connessione delle applicazioni con più installazioni vengono considerati più dettagliatamente in [Coesistenza di gestori code a più installazioni su AIX, Linux, and Windows](#).

Per ulteriori informazioni su come caricare le librerie IBM MQ , consultare [“Caricamento delle librerie IBM MQ”](#) a pagina 473.

Supporto e restrizioni

Se una delle seguenti librerie IBM MQ viene individuata e caricata, il prodotto può automaticamente caricare e utilizzare le librerie appropriate:

- Le librerie del server C
- Le librerie del server C + +
- Le librerie del server XA
- Le librerie del server COBOL
- Le librerie del server COM +
- .NET in modalità non gestita

IBM MQ carica e utilizza automaticamente anche le librerie appropriate per le applicazioni Java e JMS in modalità bind.

Esistono numerose limitazioni per le applicazioni che utilizzano più installazioni. Per ulteriori informazioni, consultare [“Limitazioni per le applicazioni che utilizzano più installazioni”](#) a pagina 476.

Concetti correlati

[“Limitazioni per le applicazioni che utilizzano più installazioni”](#) a pagina 476

Esistono delle limitazioni quando si utilizzano le librerie del server CICS , le connessioni Fast Path, gli handle dei messaggi e le uscite in un ambiente di installazione multipla.

[“Caricamento delle librerie IBM MQ”](#) a pagina 473

Quando si decide come caricare le librerie IBM MQ , è necessario considerare una serie di fattori, tra cui: l'ambiente, se è possibile modificare le applicazioni esistenti, se si desidera un'installazione primaria, dove è installato IBM MQ e se è probabile che l'ubicazione di IBM MQ cambi.

Attività correlate

[Scelta di un'installazione primaria](#)

[“Modifica dell'installazione primaria”](#) a pagina 479

È possibile utilizzare il comando **setmqinst** per impostare o annullare l'impostazione di un'installazione come installazione primaria.

[“Associazione di un gestore code a un'installazione”](#) a pagina 481

Quando si crea un gestore code, viene automaticamente associato all'installazione che ha emesso il comando **crtmqm** . Su AIX, Linux, and Windows, è possibile modificare l'installazione associata a un gestore code utilizzando il comando **setmqm** .

Caricamento delle librerie IBM MQ

Quando si decide come caricare le librerie IBM MQ , è necessario considerare una serie di fattori, tra cui: l'ambiente, se è possibile modificare le applicazioni esistenti, se si desidera un'installazione primaria, dove è installato IBM MQ e se è probabile che l'ubicazione di IBM MQ cambi.

Il modo in cui le librerie IBM MQ vengono ubicate e caricate dipende dall'ambiente di installazione:

- Sui sistemi AIX and Linux , se una copia di una versione di IBM MQ è installata nell'ubicazione predefinita, le applicazioni esistenti continuano a funzionare nello stesso modo delle versioni precedenti. Tuttavia, se le applicazioni necessitano di collegamenti simbolici in `/usr/lib`, è necessario selezionare un'installazione della versione IBM MQ come installazione primaria oppure creare manualmente i collegamenti simbolici.
- Se IBM MQ è installato in un percorso non predefinito, potrebbe essere necessario modificare le applicazioni esistenti in modo da caricare le librerie corrette.




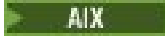
Il modo in cui le librerie IBM MQ possono essere ubicate e caricate dipende anche dal modo in cui le applicazioni esistenti sono configurate per caricare le librerie. Per ulteriori informazioni sul modo in cui è possibile caricare le librerie, consultare [“Meccanismi di caricamento della libreria del sistema operativo”](#) a pagina 475.

In modo ottimale, verificare che la libreria IBM MQ , caricata dal sistema operativo, sia quella a cui è associato il gestore code.

I metodi per caricare le librerie IBM MQ variano in base alla piattaforma e ciascun metodo presenta vantaggi e svantaggi.

Tabella 29. Vantaggi e svantaggi delle opzioni per il caricamento delle librerie			
Piattaforma	Opzione	Vantaggi	Inconvenienti
<p>Linux</p> <p>AIX</p> <p>Sistemi AIX and Linux</p>	<p>Impostare o modificare il percorso di ricerca runtime integrato (RPath) dell'applicazione.</p> <p>Questa opzione richiede di ricompilare e collegare l'applicazione. Per ulteriori informazioni relative alla compilazione e al collegamento delle applicazioni, consultare Creazione di un'applicazione procedurale.</p>	<ul style="list-style-type: none"> L'ambito della modifica è chiaro. 	<ul style="list-style-type: none"> È necessario essere in grado di ricompilare e collegare l'applicazione. Se l'ubicazione di IBM MQ cambia, è necessario modificare RPath.
<p>Sistemi AIX and Linux</p>	<p>Impostare la variabile di ambiente <code>LD_LIBRARY_PATH</code> , utilizzando <code>setmqenvo crtmqenv</code>, con l'opzione <code>-k</code> o <code>-l</code> . (</p> <p>AIX Su AIX, questa variabile di ambiente è <code>LIBPATH</code></p>	<ul style="list-style-type: none"> Non sono richieste modifiche alle applicazioni esistenti. Sovrascrive RPath integrati in un'applicazione. È facile modificare la variabile se l'ubicazione di IBM MQ cambia. 	<ul style="list-style-type: none"> Le applicazioni <code>setuid</code> e <code>setgid</code> , o le applicazioni create in altri modi, potrebbero ignorare <code>LD_LIBRARY_PATH</code> per ragioni di sicurezza. Specifico dell'ambiente, deve essere impostato in ogni ambiente in cui viene eseguita l'applicazione. Possibile impatto su altre applicazioni che si basano su <code>LD_LIBRARY_PATH</code>. Linux: il compilatore utilizzato per creare l'applicazione potrebbe disabilitare l'utilizzo di <code>LD_LIBRARY_PATH</code>. Per ulteriori informazioni, consultare Considerazioni sul collegamento di runtime per Linux .

Tabella 29. Vantaggi e svantaggi delle opzioni per il caricamento delle librerie (Continua)

Piattaforma	Opzione	Vantaggi	Inconvenienti
 Sistemi Windows	Impostare la variabile PATH utilizzando <code>setmqo crtmqenv</code> .	<ul style="list-style-type: none"> Nessuna modifica richiesta per le applicazioni esistenti. È facile modificare la variabile se l'ubicazione di IBM MQ cambia. 	<ul style="list-style-type: none"> Specifico dell'ambiente, deve essere impostato in ogni ambiente in cui viene eseguita l'applicazione. Possibile impatto su altre applicazioni.
 Sistemi AIX, Linux, and Windows	Impostare l'installazione primaria su un'installazione IBM MQo successiva. Consultare “Modifica dell'installazione primaria” a pagina 479. Per ulteriori informazioni sull'installazione primaria, consultare Scelta di un'installazione primaria .	<ul style="list-style-type: none"> Nessuna modifica richiesta per le applicazioni esistenti. Facile modificare l'installazione primaria se l'ubicazione di IBM MQ cambia. Fornisce un comportamento simile a quello delle precedenti versioni di IBM MQ. 	<ul style="list-style-type: none">   AIX and Linux: non funziona se <code>/usr/lib</code> non è nel percorso di ricerca predefinito.

Considerazioni sul caricamento della libreria per Linux

Linux

Le applicazioni compilate utilizzando alcune versioni di gcc, ad esempio, la versione 3.2.x, possono avere un RPath integrato che non può essere sovrascritto utilizzando la variabile di ambiente `LD_LIBRARY_PATH`. È possibile determinare se un'applicazione è interessata utilizzando il comando `readelf -d applicationName`. L'RPath non può essere sovrascritto se il simbolo RPATH è presente e il simbolo RUNPATH non è presente.

Meccanismi di caricamento della libreria del sistema operativo

Sui sistemi Windows, vengono ricercati diversi indirizzi per trovare le librerie:

- La directory da cui è caricata l'applicazione.
- La directory corrente.
- Le directory nella variabile di ambiente `PATH`, sia la variabile `PATH` globale che la variabile `PATH` dell'utente corrente.

Linux AIX

Sui sistemi AIX and Linux, esistono diversi metodi che potrebbero essere stati utilizzati per individuare le librerie da caricare:

- Utilizzando la variabile d'ambiente `LD_LIBRARY_PATH` (anche `LIBPATH` su AIX). Se questa variabile è impostata, definisce una serie di directory in cui vengono ricercate le librerie IBM MQ richieste. Se in queste directory vengono trovate delle librerie, esse vengono utilizzate al posto di qualsiasi libreria che potrebbe essere trovata utilizzando gli altri metodi.
- Utilizzo di un percorso di ricerca integrato (RPath). L'applicazione potrebbe contenere una serie di directory in cui ricercare le librerie IBM MQ. Se `LD_LIBRARY_PATH` non è impostato o se le librerie richieste non sono state trovate utilizzando la variabile, le librerie vengono ricercate in RPath. Se le applicazioni esistenti utilizzano un RPath, ma non è possibile ricompilare e collegare l'applicazione, è necessario installare IBM MQ nell'ubicazione predefinita o utilizzare un altro metodo per trovare le librerie.

- Viene utilizzato il percorso libreria predefinito. Se le librerie IBM MQ non vengono trovate dopo la ricerca della variabile `LD_LIBRARY_PATH` e delle ubicazioni RPath, viene ricercato il percorso della libreria predefinito. Generalmente, questo percorso contiene `/usr/lib` o `/usr/lib64`. Se le librerie non vengono trovate dopo la ricerca nel percorso libreria predefinito, l'applicazione non riesce ad avviarsi a causa di dipendenze mancanti.

È possibile utilizzare i meccanismi del sistema operativo per verificare se le applicazioni dispongono di un percorso di ricerca integrato. Ad esempio:

-  AIX: `dump`
-  Linux: `readelf`

Concetti correlati

[“Limitazioni per le applicazioni che utilizzano più installazioni” a pagina 476](#)

Esistono delle limitazioni quando si utilizzano le librerie del server CICS , le connessioni Fast Path, gli handle dei messaggi e le uscite in un ambiente di installazione multipla.

[“Connessione di applicazioni in un ambiente di installazione multiplo” a pagina 472](#)

Sui sistemi AIX, Linux, and Windows , se le librerie IBM MQ vengono caricate, IBM MQ utilizza automaticamente le librerie appropriate senza che sia necessario eseguire ulteriori operazioni. IBM MQ utilizza le librerie dell'installazione associate con il gestore code a cui si connette l'applicazione.

Attività correlate

[Scelta di un'installazione primaria](#)

[“Modifica dell'installazione primaria” a pagina 479](#)

È possibile utilizzare il comando `setmqinst` per impostare o annullare l'impostazione di un'installazione come installazione primaria.

[“Associazione di un gestore code a un'installazione” a pagina 481](#)

Quando si crea un gestore code, viene automaticamente associato all'installazione che ha emesso il comando `crtmqm` . Su AIX, Linux, and Windows , è possibile modificare l'installazione associata a un gestore code utilizzando il comando `setmqm` .

Limitazioni per le applicazioni che utilizzano più installazioni

Esistono delle limitazioni quando si utilizzano le librerie del server CICS , le connessioni Fast Path, gli handle dei messaggi e le uscite in un ambiente di installazione multipla.

Librerie server CICS

Se si utilizzano le librerie del server CICS , IBM MQ non seleziona automaticamente il livello di libreria corretto. È necessario compilare e collegare le applicazioni con il livello di libreria appropriato per il gestore code a cui si connette l'applicazione. Per ulteriori informazioni, vedi [Creazione di librerie da utilizzare con TXSeries for Multiplatforms versione 5](#).

Handle dei messaggi

Gli handle dei messaggi che utilizzano il valore speciale di `MQHC_UNASSOCIATED_HCONN` sono limitati per l'utilizzo con la prima installazione caricata in un processo. Se l'handle del messaggio non può essere utilizzato da una particolare installazione, viene restituito il codice di errore `MQRC_HMSG_NOT_AVAILABLE` .

Questa limitazione influenza le proprietà del messaggio. Non è possibile utilizzare gli handle del messaggio per richiamare le proprietà del messaggio da un gestore code su un'installazione e inserirle in un gestore code su un'installazione diversa. Per ulteriori informazioni sugli handle del messaggio, consultare [MQCRTMH - Crea handle del messaggio](#).

Uscite

In un ambiente di installazione multipla, le uscite esistenti devono essere aggiornate per essere utilizzate con installazioni IBM MQ . Le uscite di conversione dati generate utilizzando il comando **crtmqcvx** devono essere rigenerate utilizzando il comando aggiornato.

Tutte le uscite devono essere scritte utilizzando la struttura MQIEP , non possono utilizzare un RPATH integrato per individuare le librerie IBM MQ e non possono collegarsi alle librerie IBM MQ . Per ulteriori informazioni, vedere [Scrittura di uscite e servizi installabili su AIX, Linux, and Windows](#) .

Accesso rapido

Su un server con più installazioni, le applicazioni che utilizzano una connessione di percorso rapido a IBM MQ devono seguire queste regole:

1. Il gestore code deve essere associato alla stessa installazione da cui l'applicazione ha caricato le librerie di runtime di IBM MQ. L'applicazione non deve utilizzare una connessione di accesso rapido a un gestore code associato a un'installazione differente. Un tentativo di eseguire la connessione produce un errore e il codice motivo MQRC_INSTALLATION_MISMATCH.
2. Una connessione non ad accesso rapido a un gestore code associato alla stessa installazione da cui l'applicazione ha caricato le librerie di runtime di IBM MQ impedisce all'applicazione di stabilire una connessione ad accesso rapido, a meno che non sia soddisfatta una di queste condizioni:
 - L'applicazione converte la sua prima connessione a un gestore code associato alla stessa installazione in una connessione di accesso rapido.
 - La variabile di ambiente, AMQ_SINGLE_INSTALLATION è impostata.
3. La connessione di un percorso non rapido a un gestore code associato a un'installazione di IBM MQ non ha alcun effetto sul fatto che un'applicazione possa connettersi o meno al percorso rapido.

Con AMQ_SINGLE_INSTALLATION impostato, è possibile convertire qualsiasi connessione a un gestore code in una connessione di accesso rapido. In caso contrario, si applicano quasi tutte le stesse limitazioni:

- L'installazione deve essere la stessa da cui sono state caricate le librerie di runtime di IBM MQ.
- Ogni connessione sullo stesso processo deve avvenire nella stessa installazione. Se si tenta di connettersi a un gestore code associato a un'installazione differente, la connessione non riesce con codice motivo MQRC_INSTALLATION_MISMATCH. Nota: con AMQ_SINGLE_INSTALLATION impostato, questa limitazione si applica a tutte le connessioni e non solo alle connessioni di accesso rapido.
- Connettere un solo gestore code con le connessioni di accesso rapido.

Riferimenti correlati

[MQCONN - Gestore code di connessione \(esteso\)](#)

Struttura MQIEP

[2583 \(0A17\) \(RC2583\): MQRC_INSTALLATION_MISMATCH](#)

[2587 \(0A1B\) \(RC2587\): MQRC_HMSG_NOT_AVAILABLE](#)

[2590 \(0A1E\) \(RC2590\): MQRC_FASTPATH_NOT_AVAILABLE](#)

Connessione di applicazioni di .NET in un ambiente di installazione multiplo

Per impostazione predefinita, le applicazioni utilizzano gli assembly .NET dell'installazione principale. Se non esiste un'installazione primaria o non si desidera utilizzare gli assembly di installazione primari, è necessario aggiornare il file di configurazione dell'applicazione o la variabile di ambiente *DEVPATH* .

Se è presente un'installazione primaria sul sistema, gli assembly .NET e i file di politica di tale installazione vengono registrati nella GAC (global assembly cache). Gli assembly .NET per tutte le altre installazioni si trovano nel percorso di installazione di ciascuna installazione, ma gli assembly non sono registrati in GAC. Pertanto, per impostazione predefinita, le applicazioni vengono eseguite utilizzando gli assembly .NET dell'installazione primaria. È necessario aggiornare il file di configurazione dell'applicazione se si verifica uno dei seguenti casi:

- Non si dispone di un'installazione primaria.
- Non si desidera che l'applicazione utilizzi gli assembly di installazione primari.
- L'installazione principale è una versione di IBM MQ inferiore alla versione con cui è stata compilata l'applicazione.

Per informazioni su come aggiornare il file di configurazione dell'applicazione, consultare [“Connessione delle applicazioni di .NET utilizzando il file di configurazione dell'applicazione”](#) a pagina 478.

Devi aggiornare la variabile di ambiente *DEVPATH* se il seguente caso è true:

- Si desidera che l'applicazione utilizzi gli assembly da un'installazione non primaria, ma l'installazione primaria è alla stessa versione dell'installazione non primaria.

Per ulteriori informazioni su come aggiornare la variabile *DEVPATH* , consultare [“Connessione di applicazioni .NET utilizzando DEVPATH”](#) a pagina 479.

Connessione delle applicazioni di .NET utilizzando il file di configurazione dell'applicazione

All'interno del file di configurazione dell'applicazione, è necessario impostare varie tag per reindirizzare le applicazioni per utilizzare gli assembly che non provengono dall'installazione principale.

La seguente tabella mostra le specifiche modifiche che devono essere apportate al file di configurazione dell'applicazione per permettere alle applicazioni .NET di connettersi utilizzando particolari assembly:

<i>Tabella 30. Configurazione delle applicazioni per l'utilizzo di particolari assembly</i>		
	Applicazioni compilate con una versione precedente di IBM MQ	Applicazioni compilate con una versione successiva di IBM MQ
Per eseguire un'applicazione con un'installazione primaria IBM MQ di una versione successiva. (assembly di versioni successive in GAC):	Nessuna modifica necessaria	Nessuna modifica necessaria
Per eseguire un'applicazione con un'installazione primaria IBM MQ di una versione precedente. (assembly di versioni precedenti in GAC):	Nessuna modifica necessaria	Nel file di configurazione dell'applicazione: <ul style="list-style-type: none"> • Utilizzare la tag <i>bindingRedirect</i> per indicare l'utilizzo della versione precedente degli assembly presenti in GAC
Per eseguire un'applicazione con una versione successiva di installazione non primaria IBM MQ . (assembly versione successiva nella cartella di installazione):	Nel file di configurazione dell'applicazione: <ul style="list-style-type: none"> • Utilizzare la tag <i>codebase</i> per puntare all'ubicazione degli assembly delle versioni più recenti • Utilizzare la tag <i>bindingRedirect</i> per indicare l'utilizzo degli assembly delle versioni più recenti 	Nel file di configurazione dell'applicazione: <ul style="list-style-type: none"> • Utilizzare la tag <i>codebase</i> per puntare all'ubicazione degli assembly delle versioni più recenti

Tabella 30. Configurazione delle applicazioni per l'utilizzo di particolari assembly (Continua)

	Applicazioni compilate con una versione precedente di IBM MQ	Applicazioni compilate con una versione successiva di IBM MQ
Per eseguire un'applicazione con una versione precedente dell'installazione non primaria IBM MQ . (assembly di versioni precedenti nella cartella di installazione):	<p>Nel file di configurazione dell'applicazione:</p> <ul style="list-style-type: none"> Utilizzare la tag <i>codebase</i> per puntare all'ubicazione degli assembly della versione precedente Includere la tag <i>publisherpolicy Apply=no</i> 	<p>Nel file di configurazione dell'applicazione:</p> <ul style="list-style-type: none"> Utilizzare la tag <i>codebase</i> per puntare all'ubicazione degli assembly della versione precedente Utilizzare la tag <i>bindingRedirect</i> per indicare l'utilizzo degli assembly della versione precedente Includere la tag <i>publisherpolicy Apply=no</i>

Un file di configurazione dell'applicazione di esempio `NonPrimaryRedirect.config` viene fornito nella cartella `MQ_INSTALLATION_PATH\tools\dotnet\samples\base`. Questo file può essere modificato con il percorso di installazione IBM MQ di qualsiasi installazione non primaria. Il file può anche essere incluso direttamente in altri file di configurazione utilizzando la tag *linkedConfiguration* . Vengono forniti esempi per `nmqsget.exe.config` e `nmqsput.exe.config`. Entrambi gli esempi utilizzano la tag *linkedConfiguration* e includono il file `NonPrimaryRedirect.config`.

Connessione di applicazioni .NET utilizzando DEVPATH

Puoi trovare gli assembly utilizzando la variabile di ambiente `DEVPATH` . Gli assembly specificati dalla variabile `DEVPATH` vengono utilizzati di preferenza rispetto a qualsiasi assembly nel GAC. Per ulteriori informazioni su quando utilizzare questa variabile, consultare la documentazione Microsoft appropriata in `DEVPATH` .

Per trovare gli assembly utilizzando la variabile di ambiente `DEVPATH` , è necessario impostare la variabile `DEVPATH` sulla cartella che contiene gli assembly che si desidera utilizzare. Quindi, è necessario aggiornare il file di configurazione dell'applicazione e aggiungere le seguenti informazioni di configurazione runtime:

```
<configuration>
<runtime>
<developmentMode developerInstallation="true" />
</runtime>
</configuration>
```

Concetti correlati

[“Connessione di applicazioni in un ambiente di installazione multiplo” a pagina 472](#)

Sui sistemi AIX, Linux, and Windows , se le librerie IBM MQ vengono caricate, IBM MQ utilizza automaticamente le librerie appropriate senza che sia necessario eseguire ulteriori operazioni. IBM MQ utilizza le librerie dell'installazione associate con il gestore code a cui si connette l'applicazione.

[più installazioni](#)

Attività correlate

[Scelta di un'installazione primaria](#)

[Utilizzo di .NET](#)

ALW

Modifica dell'installazione primaria

È possibile utilizzare il comando `setmqinst` per impostare o annullare l'impostazione di un'installazione come installazione primaria.

Informazioni su questa attività

Questa attività si applica a AIX, Linux, and Windows.

L'installazione primaria è l'installazione a cui fanno riferimento le ubicazioni di sistema richieste. Per ulteriori informazioni sull'installazione primaria e considerazioni sulla scelta dell'installazione primaria, consultare [Scelta di un'installazione primaria](#).

Windows Durante il processo di installazione su Windows, è possibile specificare che l'installazione deve essere l'installazione primaria.

Linux **AIX** Su sistemi AIX and Linux , è necessario immettere un comando **setmqinst** dopo l'installazione per impostare l'installazione come primaria.

Procedura

- Per impostare un'installazione come installazione primaria, completare la seguente procedura:
 - a) Verificare se un'installazione è già l'installazione primaria immettendo il seguente comando:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

dove *MQ_INSTALLATION_PATH* è il percorso di installazione di un'installazione IBM MQ .

- b) Se un'installazione IBM MQ esistente è impostata come installazione primaria, [annullarla](#) prima di continuare con il passo successivo.

- c) Assicurarsi di essere collegati con l'autorizzazione appropriata:

– **Linux** **AIX** Come root su AIX and Linux.

– **Windows** Come membro del gruppo Amministratori su sistemi Windows .

- d) Immettere uno dei comandi seguenti:

– Per impostare l'installazione primaria utilizzando il percorso dell'installazione che si desidera sia l'installazione primaria:

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

– Per impostare l'installazione primaria utilizzando il nome dell'installazione che si desidera sia l'installazione primaria:

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

- e) **Windows**
Su sistemi Windows , riavviare il sistema.

- Per annullare l'impostazione di un'installazione come installazione primaria, completare la seguente procedura:

- a) Verificare quale installazione è quella primaria immettendo il seguente comando:

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

dove *MQ_INSTALLATION_PATH* è il percorso di installazione di un'installazione IBM MQ .

- b) Assicurarsi di essere collegati con l'autorizzazione appropriata:

– **Linux** **AIX** Come root su AIX and Linux.

– **Windows** Come membro del gruppo Amministratori su sistemi Windows .

- Immettere uno dei comandi seguenti:

- Per annullare l'impostazione dell'installazione primaria utilizzando il percorso dell'installazione che non si desidera più sia l'installazione primaria:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

- Per annullare l'impostazione dell'installazione primaria utilizzando il nome dell'installazione che non si desidera più sia l'installazione primaria:

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

Attività correlate

[Disinstallazione, aggiornamento e manutenzione dell'installazione primaria](#)

[Scelta di un nome di installazione](#)

Riferimenti correlati

[Funzioni che possono essere utilizzate solo con l'installazione primaria su Windows](#)

[La libreria esterna e il comando di controllo si collegano all'installazione primaria su AIX and Linux](#)
[setmqinst](#)

ALW

Associazione di un gestore code a un'installazione

Quando si crea un gestore code, viene automaticamente associato all'installazione che ha emesso il comando **crtmqm**. Su AIX, Linux, and Windows, è possibile modificare l'installazione associata a un gestore code utilizzando il comando **setmqm**.

Informazioni su questa attività

L'installazione che un gestore code è associato limita tale gestore code in modo che possa essere gestito solo dai comandi di tale installazione. Esistono tre eccezioni chiave:

- **setmqm** modifica l'installazione associata al gestore code. Questo comando deve essere immesso dall'installazione che si desidera associare al gestore code, non dall'installazione a cui è attualmente associato il gestore code. Il nome di installazione specificato dal comando **setmqm** deve corrispondere all'installazione da cui viene emesso il comando.
- **strmqm** deve essere emesso dall'installazione associata al gestore code.
- **dspm** visualizza informazioni su tutti i gestori code su un sistema, non solo su quelli associati alla stessa installazione del comando **dspm**. Il comando `dspm -o installation` visualizza informazioni sui gestori code associati a quali installazioni.

Per gli ambienti HA, il comando **addmqinf** associa automaticamente il gestore code all'installazione da cui viene emesso il comando **addmqinf**. Finché il comando **strmqm** viene immesso dalla stessa installazione del comando **addmqinf**, non è necessaria alcuna ulteriore configurazione. Per avviare il gestore code utilizzando un'installazione differente, è necessario prima modificare l'installazione associata mediante il comando **setmqm**.

Quando si desidera associare un gestore code a un'installazione, è possibile utilizzare il comando **setmqm** nei seguenti modi:

- Spostamento di singoli gestori code tra versioni equivalenti di IBM MQ. Ad esempio, lo spostamento di un gestore code da una prova a un sistema di produzione.
- Migrazione di singoli gestori code da una versione precedente di IBM MQ a una versione più recente di IBM MQ. La migrazione dei gestori code tra versioni ha diverse implicazioni di cui è necessario essere consapevoli. Per ulteriori informazioni sulla migrazione, consultare [Manutenzione e migrazione](#).

Procedura

1. Arrestare il gestore code utilizzando il comando **endmqm** dall'installazione attualmente associata al gestore code.

2. Associare il gestore code a un'altra installazione utilizzando il comando **setmqm** da tale installazione.
Ad esempio, per impostare il gestore code QMB in modo che sia associato a un'installazione con il nome `Installation2`, immettere il comando seguente da `Installation2`:

```
MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
```

dove `MQ_INSTALLATION_PATH` è il percorso in cui è installato `Installation2`.

3. Avviare il gestore code utilizzando il comando **strmqm** dall'installazione ora associata al gestore code.
Questo comando esegue la migrazione del gestore code necessaria e consente al gestore code di essere pronto per l'uso.

Operazioni successive

Se l'installazione a cui è associato un gestore code è stata eliminata o se le informazioni sullo stato del gestore code non sono disponibili, il comando **setmqm** non riesce ad associare il gestore code ad un'altra installazione. In questa situazione, effettuare quanto segue:

1. Utilizzare il comando **dspmqrinst** per visualizzare le altre installazioni sul sistema.
2. Modificare manualmente il campo `InstallationName` della sezione `QueueManager` in `mqs.ini` per specificare un'altra installazione.
3. Utilizzare il comando **dltmqm** da tale installazione per eliminare il gestore code.

Concetti correlati

[“Ricerca di installazioni di IBM MQ su un sistema” a pagina 482](#)

Se si dispone di più installazioni IBM MQ su un sistema, è possibile verificare quali versioni sono installate e dove si trovano.

[“File di configurazione IBM MQ , mqs.ini” a pagina 94](#)

Il file di configurazione IBM MQ , `mqs.ini`, contiene informazioni relative a tutti i gestori code sul nodo. Viene creato automaticamente durante l'installazione.

Attività correlate

[Scelta di un'installazione primaria](#)

Riferimenti correlati

[addmqinf](#)

[dspmqr](#)

[dspmqrinst](#)

[endmqm](#)

[setmqm](#)

[strmqm](#)

ALW

Ricerca di installazioni di IBM MQ su un sistema

Se si dispone di più installazioni IBM MQ su un sistema, è possibile verificare quali versioni sono installate e dove si trovano.

È possibile utilizzare i seguenti metodi per individuare le installazioni IBM MQ sul proprio sistema:

- Utilizzare gli strumenti di installazione della piattaforma per eseguire una query su dove è stato installato IBM MQ . Utilizzare quindi il comando **dspmqrver** da un'installazione IBM MQ . I seguenti comandi sono esempi di comandi che è possibile utilizzare per eseguire una query su dove è stato installato IBM MQ :

–  Sui sistemi AIX , è possibile utilizzare il comando **lslpp** :

```
lslpp -R ALL -l mqm.base.runtime
```

- **Linux** Sui sistemi Linux , è possibile utilizzare il comando **rpm** :

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

- **Windows** Su sistemi Windows , è possibile utilizzare il comando **wmic** . Questo comando potrebbe installare il client wmic:

```
wmic product where "(Name like 'MQ%') AND (not Name like '%bitSupport')" get Name, Version, InstallLocation
```

- **Linux** **AIX** Su sistemi AIX and Linux , immettere il seguente comando per individuare dove è stato installato IBM MQ :

```
cat /etc/opt/mqm/mqinst.ini
```

Utilizzare quindi il comando **dspmqr** da un'installazione IBM MQ .

- **Windows** Per visualizzare i dettagli delle installazioni sul sistema, su Windows a 32 bit, immettere il seguente comando:

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
```

- **Windows** Su Windows a 64 bit, immettere il seguente comando:

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
```

Riferimenti correlati

[dspmqr](#)

[dspmqrinst](#)

[più installazioni](#)

Configurazione dell'alta disponibilità, ripristino e riavvio

È possibile rendere le applicazioni altamente disponibili mantenendo la disponibilità della coda in caso di errore di un gestore code e ripristinando i messaggi dopo un errore del server o della memoria.




Informazioni su questa attività


z/OS Su z/OS, l'alta disponibilità è integrata nella piattaforma. Vedere [Code condivise e gruppi di condivisione code](#).


Multi Su Multiplatforme, è possibile migliorare la disponibilità delle applicazioni client utilizzando la riconnessione client per commutare automaticamente un client tra un gruppo di gestori code o alla nuova istanza attiva di un gestore code a più istanze dopo un malfunzionamento del gestore code. La riconnessione automatica del client non è supportata da IBM MQ classes for Java. Un gestore code a più istanze è configurato per essere eseguito come singolo gestore code su più server. Le applicazioni server vengono distribuite a questo gestore code. Se il server che esegue l'istanza attiva ha esito negativo, l'esecuzione viene automaticamente commutata in un'istanza in standby dello stesso gestore code su un server differente. Se si configurano le applicazioni del server per l'esecuzione come servizi del gestore code, queste vengono riavviate quando un'istanza in standby diventa l'istanza del gestore code in esecuzione attiva.


Un altro modo per aumentare la disponibilità delle applicazioni server su Multiplatforms è quello di distribuire le applicazioni server a più computer in un cluster di gestori code. Da IBM WebSphere MQ 7.1 in poi, il ripristino da errori del cluster riesegui le operazioni che hanno causato problemi fino a

quando i problemi non vengono risolti. Consultare [“Modifiche al ripristino da errori cluster sui server su Multiplatforms”](#) a pagina 703. È anche possibile configurare IBM MQ for Multiplatforms come parte di una soluzione di clustering specifica della piattaforma, ad esempio:

- Microsoft Server cluster
-  Cluster HA su IBM i
-   PowerHA per AIX (precedentemente HACMP su AIX) e altre soluzioni di clustering UNIX and Linux

 Su sistemi Linux, è possibile configurare i gestori code di dati replicati (RDQM) per implementare soluzioni di alta disponibilità o di ripristino di emergenza. Per l'alta disponibilità, le istanze dello stesso gestore code sono configurate su ciascun nodo in un gruppo di tre server Linux. Una delle tre istanze è l'istanza attiva. I dati del gestore code attivo vengono replicati in modo sincrono nelle altre due istanze, quindi una di queste istanze può assumere il controllo in caso di errore. Per il ripristino di emergenza, un gestore code viene eseguito su un nodo primario su un sito, con un'istanza secondaria di tale gestore code ubicata su un nodo di ripristino su un sito differente. I dati vengono replicati tra l'istanza primaria e l'istanza secondaria, e se il nodo primario viene perso per qualche motivo, l'istanza secondaria può essere creata nell'istanza primaria e avviata.

 La HA nativa è una soluzione di alta disponibilità rivolta ai contenitori. La HA nativa utilizza la replica del log per mantenere aggiornate tre istanze di un gestore code in esecuzione su nodi differenti. Un'istanza è attiva in qualsiasi momento ed elabora i messaggi. Il gestore code attivo invia i propri aggiornamenti log alle altre due istanze per mantenerli aggiornati. Se l'istanza attiva ha esito negativo, una delle istanze di replica assume automaticamente il ruolo attivo.

 Un'altra opzione per una soluzione di ripristino di emergenza o alta disponibilità è quella di distribuire una coppia di dispositivi IBM MQ. Vedi [High Availability](#) e [Disaster Recovery](#) nella documentazione di IBM MQ Appliance.

Un sistema di messaggistica garantisce che i messaggi immessi nel sistema vengano consegnati alla loro destinazione. IBM MQ può tracciare l'instradamento di un messaggio mentre si sposta da un gestore code all'altro utilizzando il comando **dspmqrte**. Se un sistema ha esito negativo, i messaggi possono essere ripristinati in vari modi a seconda del tipo di errore e del modo in cui un sistema è configurato. IBM MQ conserva i log di recupero delle attività dei gestori code che gestiscono la ricezione, trasmissione e consegna dei messaggi. Utilizza questi log per tre tipi di ripristino:

1. *Riavviare il recupero*, quando si arresta IBM MQ in modo pianificato.
2. *Ripristino da errore*, quando un errore si arresta IBM MQ.
3. *Ripristino dei supporti*, per ripristinare gli oggetti danneggiati.

In tutti i casi, il ripristino ripristina il gestore code allo stato in cui si trovava quando il gestore code è stato arrestato, ad eccezione del fatto che le transazioni in corso vengono sottoposte a rollback, rimuovendo dalle code gli aggiornamenti in corso al momento dell'arresto del gestore code. Il ripristino ripristina tutti i messaggi persistenti; i messaggi non persistenti potrebbero essere persi durante il processo.



Avvertenza: Non è possibile spostare i log di ripristino su un sistema operativo differente.

Riconnessione automatica del client

È possibile riconnettere automaticamente le proprie applicazioni client, senza scrivere alcun codice aggiuntivo, configurando un certo numero di componenti.

La riconnessione automatica del client è *in linea*. La connessione viene ripristinata automaticamente in qualsiasi punto del programma applicativo client e vengono ripristinati anche tutti gli handle per aprire gli oggetti.

Al contrario, la riconnessione manuale richiede l'applicazione client per ricreare una connessione utilizzando MQCONN o MQCONNX e riaprire gli oggetti. La riconnessione automatica del client è adatta a molte ma non a tutte le applicazioni client.

Tabella 31 a pagina 485 elenca il primo release del supporto client IBM MQ che deve essere installato su una workstation client. È necessario aggiornare le stazioni di lavoro client a uno di questi livelli affinché un'applicazione possa utilizzare la riconnessione client automatica. La Tabella 32 a pagina 486 elenca altri requisiti per abilitare la riconnessione automatica del client.

Con l'accesso del programma alle opzioni di riconnessione, un'applicazione client può impostare le opzioni di riconnessione. Ad eccezione dei client JMS e XMS, se un'applicazione client ha accesso alle opzioni di riconnessione, può anche creare un gestore eventi per gestire gli eventi di riconnessione.

Un'applicazione client esistente potrebbe beneficiare del supporto di riconnessione, senza ricompilazione e collegamento:

- Per un client non JMS, impostare la `mqclient.ini` variabile di ambiente `DefRecon` per impostare opzioni di riconnessione. Utilizzare una CCDT per connettersi a un gestore code. Se il client deve connettersi a un gestore code a più istanze, fornire gli indirizzi di rete delle istanze del gestore code attivo e in standby in CCDT. Per un gestore code di dati replicati o un gestore code HA su IBM MQ Appliance, è possibile specificare un indirizzo IP mobile utilizzato sia dai gestori code attivi che da quelli in standby per semplificare la configurazione.
- Per un client JMS, impostare le opzioni di riconnessione nella configurazione della produzione connessioni. Durante l'esecuzione nel contenitore EJB di un server Java EE, gli MDB possono riconnettersi a IBM MQ utilizzando il meccanismo di riconnessione fornito dalle specifiche di attivazione dell'adattatore di risorse IBM MQ (o le porte listener se in esecuzione in WebSphere Application Server). Tuttavia, se l'applicazione non è un MDB (o è in esecuzione nel contenitore Web), l'applicazione deve implementare la propria logica di riconnessione poiché la riconnessione automatica del client non è supportata in questo scenario. L'adattatore di risorse IBM MQ fornisce questa capacità di riconnessione per la consegna dei messaggi ai bean basati sui messaggi, ma altri elementi Java EE come i servlet devono implementare la propria riconnessione.

Nota: La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

Interfaccia Client	Client	Accesso del programma alle opzioni di riconnessione	Supporto riconnessione
API di messaggistica	C, C++, COBOL, Unmanaged Visual Basic XMS (Unmanaged XMS on Windows)	7.0.1	7.0.1
	JMS (contenitore client JSE e Java EE e contenitori gestiti)	7.0.1.3	7.0.1.3
	IBM MQ classes for Java	Non supportato	Non supportato
	Client XMS e .NET gestiti: C#, Visual Basic,	7.1	7.1

Tabella 31. Client supportati (Continua)

Interfaccia Client	Client	Accesso del programma alle opzioni di riconnessione	Supporto riconnessione
Altre API	Windows Communication Foundation (¹ non gestito)	Non supportato	7.0.1
	Windows Communication Foundation (gestito ¹)	Non supportato	Non supportato
	Asse 1	Non supportato	Non supportato
	Asse 2	Non supportato	7.0.1.3
	HTTP (web 2.0)	Non supportato	7.0.1.3

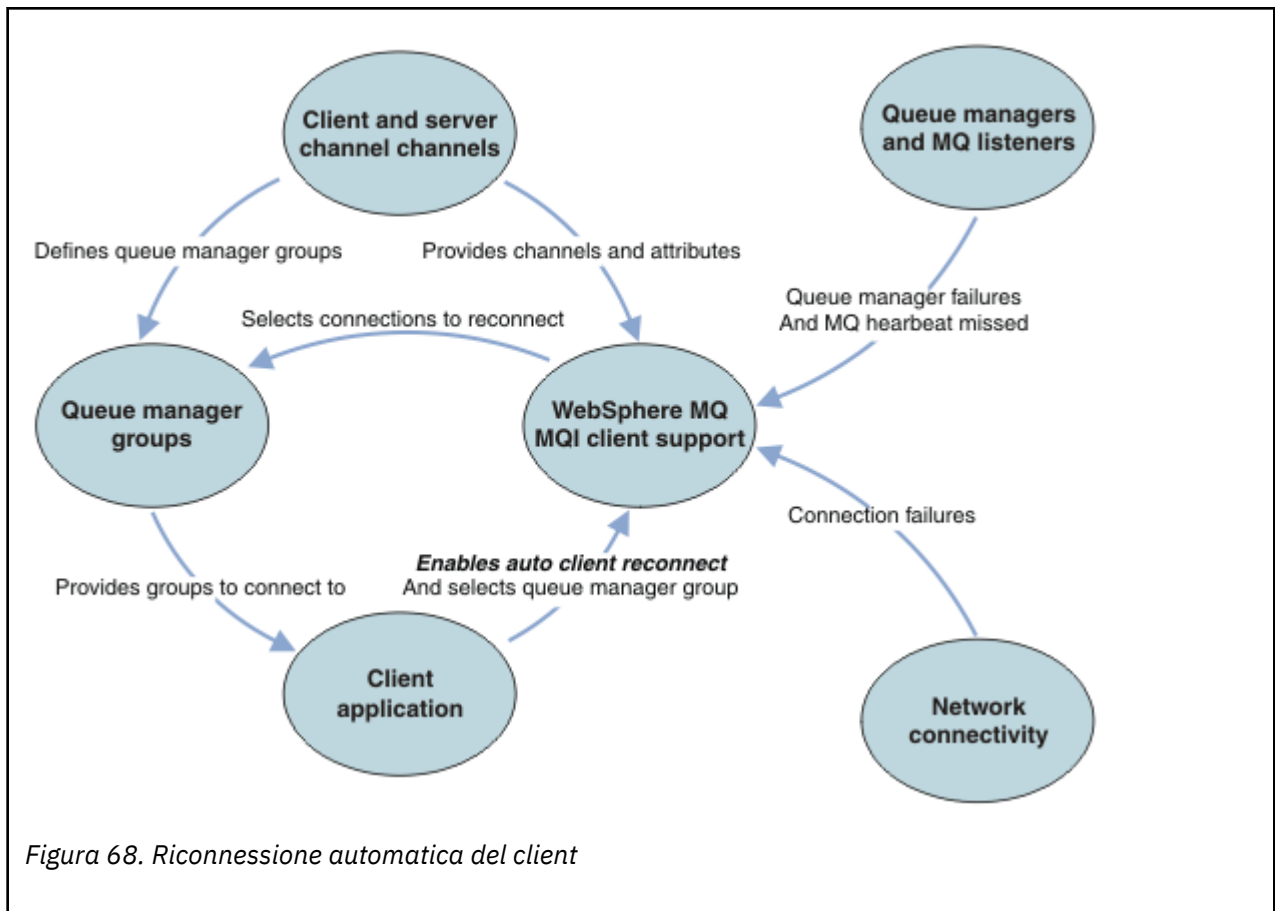
1. Impostare la modalità gestita o non gestita nella configurazione del bind WCF.

La riconnessione automatica ha i seguenti requisiti di configurazione:

Tabella 32. Requisiti di configurazione della riconnessione automatica

Componente	Requisiti	Effetto del mancato rispetto dei requisiti
IBM MQ MQI client installazione	Vedere Tabella 31 a pagina 485	MQRC_OPTIONS_ERROR
IBM MQ Installazione server	Livello 7.0.1	MQRC_OPTIONS_ERROR
Canale	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
ambiente applicativo	Deve essere con thread	MQRC_ENVIRONMENT_ERROR
MQI	Uno tra: <ul style="list-style-type: none"> MQCONNX con MQCNO Opzioni impostate su MQCNO_RECONNECT o MQCNO_RECONNECT_Q_MGR. Defrecon=YES QMGR in mqclient.ini In JMS impostare la proprietà CLIENTRECONNECTOPTIONS del factory di connessione. 	MQCC_FAILED quando una connessione è interrotta o il gestore code termina o non riesce.

La Figura 68 a pagina 487 mostra le interazioni principali tra i componenti coinvolti nella riconnessione client.



Applicazione client

L'applicazione client è un IBM MQ MQI client. Per dettagli sulla riconnessione automatica del client per un client JMS, consultare [Utilizzo della riconnessione automatica del client JMS](#).

- Per impostazione predefinita, i client non vengono riconnessi automaticamente. Abilitare la riconnessione automatica del client impostando l'opzione MQCONNX MQCNO MQCNO_RECONNECT o MQCNO_RECONNECT_Q_MGR.
- Molte applicazioni sono scritte in modo che siano in grado di sfruttare la riconnessione automatica senza alcuna codifica aggiuntiva. Abilitare la riconnessione automatica per programmi esistenti, senza apportare alcuna modifica di codifica, impostando l'attributo DefRecon nella stanza dei canali del file di configurazione mqclient.ini.
- Utilizzare una delle seguenti tre opzioni:
 1. Modificare il programma in modo che la logica non sia influenzata dalla riconnessione. Ad esempio, potrebbe essere necessario emettere chiamate MQI all'interno del punto di sincronizzazione e inoltrare nuovamente le transazioni di cui è stato eseguito il backout. I consumer asincroni devono controllare se sono stati 'sospesi' se viene eseguito il backout di una transazione.
 2. Aggiungere un gestore eventi per rilevare la riconnessione e ripristinare lo stato dell'applicazione client quando la connessione viene ristabilita.
 3. Non abilitare la riconnessione automatica: disconnettere il client ed emettere una nuova chiamata MQI MQCONN o MQCONNX per individuare un'altra istanza del gestore code in esecuzione nello stesso gruppo di gestori code.

Per ulteriori dettagli su queste tre opzioni, consultare [“Ripristino applicazione”](#) a pagina 581.

- La riconnessione a un gestore code con lo stesso nome non garantisce la riconnessione alla stessa istanza di un gestore code.

Utilizzare un'opzione MQ MQCNO_RECONNECT_Q_MGR per riconnettersi a un'istanza dello stesso gestore code.

- Un client può registrare un gestore eventi in modo che possa essere informato dello stato di riconnessione. Il MQHCONN passato nel gestore eventi non può essere utilizzato. Vengono forniti i seguenti codici di errore:

MQRC_RECONNECTING

La connessione non è riuscita e il sistema sta tentando di riconnettersi. Si ricevono più eventi MQRC_RECONNECTING se vengono effettuati più tentativi di riconnessione.

MQRC_RECONNECTED

La riconnessione è stata effettuata e tutti gli handle sono stati ristabiliti correttamente.

MQRC_RECONNECT_NON RIUSCITO

La riconnessione non è riuscita.

MQRC_RECONNECT_QMID_MISMATCH

Una connessione ricollegabile ha specificato MQCNO_RECONNECT_Q_MGR e la connessione ha tentato di riconnettersi a un gestore code differente.

MQRC_RECONNECT_Q_MGR_REQD

Un'opzione, come MQMO_MATCH_MSG_TOKEN in una chiamata MQGET , è stata specificata nel programma del client che richiede la riconnessione allo stesso gestore code.

- Un client ricollegabile è in grado di riconnettersi automaticamente solo dopo la connessione. In altre parole, la chiamata MQCONNX non viene ritentata se ha esito negativo. Ad esempio, se si riceve il codice di ritorno 2543 - MQRC_STANDBY_Q_MGR da MQCONNX, emettere nuovamente la chiamata dopo un breve ritardo.

MQRC_RECONNECT_INCOMPATIBILE

Questo codice motivo viene restituito quando l'applicazione tenta di utilizzare MQPMO_LOGICAL_ORDER (con MQPUT e MQPUT1) o MQGMO_LOGICAL_ORDER (con MQGET) quando sono impostate le opzioni di riconnessione. Il motivo per restituire il codice di errore è assicurarsi che le applicazioni non utilizzino mai la riconnessione in tali casi.

MQRC_CALL_INTERROTTO

Questo codice motivo viene restituito quando la connessione si interrompe durante l'esecuzione della chiamata Commit e il client si riconnette. Un MQPUT di un messaggio persistente esterno al punto di sincronizzazione determina anche la restituzione dello stesso codice di errore all'applicazione.

Gestori code ad alta disponibilità

I gestori code ad alta disponibilità hanno un'istanza attiva e una o più istanze in standby di un gestore code. Il gestore code attivo è sincronizzato con i gestori code in standby, in modo che uno standby possa eseguire automaticamente il takeover se l'istanza attiva ha esito negativo. Esistono diverse soluzioni per fornire gestori code ad elevata disponibilità, consultare [“Configurazioni HA \(High Availability\)” a pagina 495](#).

È possibile semplificare il riavvio delle applicazioni IBM MQ MQI client , dopo che un gestore code ad alta disponibilità ha attivato la propria istanza in standby, utilizzando la riconnessione client automatica.

L'istanza in standby di un gestore code ad alta disponibilità è in genere a un indirizzo di rete differente rispetto all'istanza attiva. Includere gli indirizzi di rete di entrambe le istanze nella CCDT (client connection definition table). Fornire un elenco di indirizzi di rete per il parametro **CONNAME** oppure definire più righe per il gestore code in CCDT. I gestori code di dati replicati e i gestori code ad alta disponibilità di IBM MQ Appliance supportano gli indirizzi IP mobili, dove si specifica un singolo indirizzo da utilizzare con i gestori code attivi o in standby.

Gruppi gestore code


In genere, IBM MQ MQI clients si riconnette a qualsiasi gestore code in un gruppo di gestori code. A volte si desidera che un IBM MQ MQI client si riconnette solo allo stesso gestore code. Potrebbe avere un'affinità con un gestore code.

È possibile selezionare se l'applicazione client si connette sempre e si riconnette a un gestore code con lo stesso nome, allo stesso gestore code o a uno qualsiasi dei gestori code definiti con lo stesso valore QMNAME nella tabella di connessione client.

- L'attributo del nome gestore code, QMNAME, nella definizione del canale client è il nome di un gruppo di gestori code.
- Nell'applicazione client, se si imposta il valore del parametro MQCONN o MQCONNX QmgrName su un nome gestore code, il client si connette solo ai gestori code con tale nome. Se si aggiunge un asterisco (*) al nome del gestore code, il client si connette a qualsiasi gestore code nel gruppo di gestori code con lo stesso valore QMNAME . Per una spiegazione completa, consultare [Gruppi di gestori code in CCDT](#).

È possibile evitare che un client si riconnette a un gestore code differente. Impostare l'opzione MQCNO , MQCNO_RECONNECT_Q_MGR. IBM MQ MQI client ha esito negativo se si riconnette a un gestore code differente. Se si imposta l'opzione MQCNO , MQCNO_RECONNECT_Q_MGR, non includere gli altri gestori code nello stesso gruppo di gestori code. Il client restituisce un errore se il gestore code a cui si riconnette non è lo stesso gestore code a cui si è connesso.

Gruppi di condivisione code

 La riconnessione automatica del client ai gruppi di condivisione code z/OS utilizza gli stessi meccanismi di riconnessione di qualsiasi altro ambiente. Il client si riconnetterà alla stessa selezione di gestori code configurata per la connessione originale. Ad esempio, quando si utilizza la tabella di definizione del canale client, l'amministratore deve verificare che tutte le voci nella tabella, si risolvano nello stesso gruppo di condivisione code z/OS .

Definizioni di canali client e server

Le definizioni dei canali client e server definiscono i gruppi di gestori code a cui un'applicazione client può riconnettersi. Le definizioni gestiscono la selezione e la tempistica delle riconnessioni e altri fattori, come la sicurezza; consultare gli argomenti correlati. Gli attributi del canale più rilevanti da considerare per la riconnessione sono elencati in due gruppi:

Attributi di connessione client

Affinità di connessione (AFFINITY) AFFINITY

Affinità connessione.

Peso del canale client (CLNTWGHT) CLNTWGHT

Importanza del canale del client.

Nome connessione (CONNAME) CONNAME

Informazioni di connessione.

Intervallo heartbeat (HBINT) HBINT

Intervallo heartbeat. Impostare l'intervallo di heartbeat sul canale di connessione server.

Intervallo keepalive (KAINT) KAINT

Intervallo keepalive. Impostare l'intervallo keepalive sul canale di connessione server.

 Notare che KAINT si applica solo a z/OS .

Nome gestore code (QMNAME) QMNAME

È il nome del gestore code.

Attributi connessione server

Intervallo heartbeat (HBINT) HBINT

Intervallo heartbeat. Impostare l'intervallo di heartbeat sul canale di connessione client.

Intervallo keepalive (KAINT) KAINT

Intervallo keepalive. Impostare l'intervallo keepalive sul canale di connessione client.



Notare che KAIN T si applica solo a z/OS .

KAIN T è un heartbeat a livello di rete e HBINT è un heartbeat IBM MQ tra client e gestore code. L'impostazione di questi heartbeat su un tempo più breve serve a due scopi:

1. Simulando l'attività sulla connessione, il software del livello di rete responsabile della chiusura delle connessioni inattive ha meno probabilità di chiudere la connessione.
2. Se la connessione viene chiusa, il ritardo prima che venga rilevata la connessione interrotta viene abbreviato.

L'intervallo di keepalive TCP/IP predefinito è due ore. Impostare gli attributi KAIN T e HBINT su un periodo di tempo più breve. Non presumere che il normale funzionamento di una rete si adatti alle esigenze di riconnessione automatica. Ad esempio, alcuni firewall possono arrestare una connessione TCP/IP inattiva dopo soli 10 minuti.

Connettività di rete

Solo gli errori di rete trasmessi a IBM MQ MQI client dalla rete vengono gestiti dalla funzionalità di riconnessione automatica del client.

- Le riconnessioni eseguite automaticamente dal trasporto sono invisibili a IBM MQ.
- L'impostazione HBINT consente di gestire gli errori di rete invisibili a IBM MQ.

Gestori code e listener IBM MQ

La riconnessione client viene attivata per errore del server, errore del gestore code, errore di connettività di rete e da un amministratore che passa a un'altra istanza del gestore code.

- Se si utilizza un gestore code a più istanze, un'ulteriore causa della riconnessione del client si verifica quando si passa il controllo dall'istanza del gestore code attivo a un'istanza in standby.
- L'arresto di un gestore code utilizzando il comando **endmqm** predefinito non attiva la riconnessione automatica del client. Aggiungere l'opzione **-r** sul comando **endmqm** per richiedere la riconnessione automatica del client o l'opzione **-s** per il trasferimento a un'istanza del gestore code in standby dopo la chiusura.

IBM MQ MQI client supporto di riconnessione automatica

Se si utilizza il supporto di riconnessione client automatica in IBM MQ MQI client, l'applicazione client si riconnette automaticamente e continua l'elaborazione senza emettere una chiamata MQCONN o MQCONNX MQI per riconnettersi al gestore code.

- La riconnessione automatica del client viene attivata da uno dei seguenti eventi:
 - Errore del gestore code
 - Chiusura di un gestore code e specifica dell'opzione **-r**, riconnessione, nel comando **endmqm**
- Le opzioni di MQCONNX MQCNO controllano se è stata abilitata la riconnessione automatica del client. Le opzioni sono descritte in [Opzioni di riconnessione](#).
- La riconnessione automatica del client emette chiamate MQI per conto dell'applicazione per ripristinare l'handle di connessione e gli handle per altri oggetti aperti, in modo che il programma possa riprendere la normale elaborazione dopo aver elaborato gli errori MQI risultanti dalla connessione interrotta. Consultare ["Ripristino di un client riconnesso automaticamente"](#) a pagina 583.
- Se è stato scritto un programma di uscita del canale per la connessione, l'exit riceve queste chiamate MQI aggiuntive.
- È possibile registrare un gestore eventi di riconnessione, che viene attivato all'inizio e al termine della riconnessione.

Anche se il tempo di riconnessione previsto non è superiore a un minuto, la riconnessione può richiedere più tempo perché un gestore code potrebbe avere numerose risorse da gestire. Durante questo periodo, un'applicazione client potrebbe mantenere dei blocchi che non appartengono alle risorse IBM MQ .

Esiste un valore di timeout che è possibile configurare per limitare il tempo di attesa di un client per la riconnessione. Il valore (in secondi) è impostato nel file `mqclient.ini`.

```
Channels:  
MQReconnectTimeout = 1800
```

Non viene effettuato alcun tentativo di riconnessione dopo la scadenza del timeout. Quando il sistema rileva che il timeout è scaduto restituisce un errore `MQRC_RECONNECT_FAILED`.

Concetti correlati

[Client riconnettibili](#)

Attività correlate

[Arresto di un gestore code](#)

z/OS

Console message monitoring

On IBM MQ for z/OS, there are a number of information messages issued by the queue manager or channel initiator that should be considered particularly significant. These messages do not in themselves indicate a problem, but can be useful in tracking because they do indicate a potential issue which might need addressing.

The presence of these console messages might also indicate that a user application is putting a large number of messages to the page set, which might be a symptom of a larger problem:

- A problem with the user application which PUTs messages, such as an uncontrolled loop.
- A user application which GETs the messages from the queue is no longer functioning.

Console messages to monitor

The following list outlines messages which can potentially indicate larger problems. Determine if it is necessary to track these messages with system automation and provide appropriate documentation so any potential problems can be followed up effectively.

CSQI004I: csect-name CONSIDER INDEXING queue-name BY index-type FOR connection-type CONNECTION connection-name, num-msgs MESSAGES SKIPPED

- The queue manager has detected an application receiving messages by message ID or correlation ID from a queue that does not have an index defined.
- Consider establishing an index for the identified queue by altering the local queue object, *queue-name*, `INDXTYPE` attribute to have value *index-type*.

CSQI031I: csect-name THE NEW EXTENT OF PAGE SET psid HAS FORMATTED SUCCESSFULLY

- Check the curdepth of the queues allocated to this page set.
- Investigate the cause of the failure to process the messages.

CSQI041I: csect-name JOB jobname USER userid HAD ERROR ACCESSING PAGE SET psid

- Determine if the page set is allocated to the queue manager.
- Issue a **DISPLAY USAGE** command to determine the state of the page set.
- Check the queue manager joblog for additional error messages.

CSQI045I: csect-name Log RBA has reached rba. Plan a log reset

- Plan to stop the queue manager at a convenient time and reset the logs.
- If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs.

CSQI046E: csect-name Log RBA has reached rba. Perform a log reset

- Plan to stop the queue manager at a convenient time and reset the logs.
- If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs.

CSQI047E: csect-name Log RBA has reached rba. Stop queue manager and reset logs

- Stop the queue manager immediately and reset the logs.
- If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs.

CSQJ004I: ACTIVE LOG COPY *n* INACTIVE, LOG IN SINGLE MODE, ENDRBA= *ttt*

- The queue manager has activated 'single' logging mode. This is often indicative of a log offload problem.
- Issue a **DISPLAY LOG** command to determine your settings for duplexing of active and archive logs. This display also shows how many active logs need offload processing.
- Check the queue manager joblog for additional error messages

CSQJ031D: csect-name, THE LOG RBA RANGE MUST BE RESET. REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN

- Stop the queue manager and reset the logs as soon as possible and reset the logs.
- If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs.

CSQJ032E: csect-name alert-lvl - APPROACHING END OF THE LOG RBA RANGE OF *max-rba*. CURRENT LOG RBA IS *current-rba*.

- Plan to stop the queue manager and reset the logs as soon as possible.
- If your queue manager is using 6-byte log RBAs, consider converting the queue manager to use 8-byte log RBAs.

CSQJ110E: LAST COPY *n* ACTIVE LOG DATA SET IS *nnn* PERCENT FULL

- Take steps to complete other waiting offload tasks by performing a display request to determine the outstanding requests related to the log offload process. Take the necessary action to satisfy any requests, and permit offload to continue.
- Consider whether there are sufficient active log data sets. If necessary, you can add additional log data sets dynamically by using the [DEFINE LOG](#) command.

CSQJ111A: OUT OF SPACE IN ACTIVE LOG DATA SETS

- Perform a display request to ensure that there are no outstanding requests that are related to the log offload process. Take the necessary action to satisfy any requests, and permit offload to continue.
- Consider whether there are sufficient active log data sets. If necessary, you can add additional log data sets dynamically by using the [DEFINE LOG](#) command.
- If the delay was caused by the lack of a resource required for offload, the necessary resource must be made available to allow offload to complete and thus permit logging to proceed. For information about recovery from this condition, see [Archive log problems](#).

CSQJ114I: ERROR ON ARCHIVE DATA SET, OFFLOAD CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING GENERATED

- Check the queue manager joblog for additional error messages.
- Make a second copy of the archive log and update your BSDS manually.

CSQJ115E: OFFLOAD FAILED, COULD NOT ALLOCATE AN ARCHIVE DATA SET

Review the error status information of message CSQJ103E or CSQJ073E. Correct the condition that caused the data set allocation error so that, on retry, the offload can take place.

CSQJ136I: UNABLE TO ALLOCATE TAPE UNIT FOR CONNECTION-ID= *xxxx* CORRELATION-ID= *yyyyyy*, *m* ALLOCATED *n* ALLOWED

- Check the queue manager joblog for additional error messages.

CSQJ151I: *csect-name* ERROR READING RBA *rrr*, CONNECTION-ID= *xxxx* CORRELATION-ID= *yyyyyy* REASON CODE= *ccc*

- Check the queue manager joblog for additional messages.
- Issue a **DISPLAY CONN** command to determine which connection is not committing its activity.
- Ensure the application can commit its updates.

CSQJ160I: LONG-RUNNING UOW FOUND, URID= *urid* CONNECTION NAME= *name*

- Check the queue manager joblog for additional messages.
- Issue a **DISPLAY CONN** command to determine which connection is not committing its activity.
- Ensure the application can commit its updates.

CSQJ161I: UOW UNRESOLVED AFTER *n* OFFLOADS, URID= *urid* CONNECTION NAME= *name*

- Determine if the page set is allocated to the queue manager.
- Issue a **DISPLAY USAGE** command to determine the state of the page set.
- Check the queue manager joblog for additional messages.

CSQP011E: CONNECT ERROR STATUS *ret-code* FOR PAGE SET *psid*

- Check the curdepth of the queues allocated to this page set.
- Investigate the cause of the failure to process messages.

CSQP013I: *csect-name* NEW EXTENT CREATED FOR PAGE SET *psid*. NEW EXTENT WILL NOW BE FORMATTED

- Check the curdepth of the queues allocated to this page set.
- Investigate the cause of failure to process messages.
- Determine if queues need to be relocated to another page set.
- If the volume is full, determine if you need to make the page set a multi volume data set. If the page set is already multi-volume, consider adding more volumes to the storage group being used. Once more space is available retry the expansion by setting the page set **EXPAND** method to **SYSTEM**. If a retry is required, toggle **EXPAND** to **SYSTEM** and then back to your normal setting.

CSQP014E: *csect-name* EXPANSION FAILED FOR PAGE SET *psid*. FUTURE REQUESTS TO EXTEND IT WILL BE REJECTED

- Check the curdepth of the queues allocated to this page set.
- Investigate the cause of failure to process messages.
- Determine if queues need to be relocated to another page set.

CSQP016E: *csect-name* PAGE SET *psid* HAS REACHED THE MAXIMUM NUMBER OF EXTENTS. IT CANNOT BE EXTENDED AGAIN

- Check the curdepth of the queues allocated to this page set.
- Investigate the cause of failure to process messages.

CSQP017I: *csect-name* EXPANSION STARTED FOR PAGE SET *psid*

Issue DISPLAY THREAD commands to determine the state of the Units of Work in IBM MQ.

CSQP047E: Unavailable page sets can cause problems - take action to correct this situation

- Follow the system programmer response.

CSQQ008I: nn units of recovery are still in doubt in queue manager qqqq

- Investigate the state of your dead letter queue. Ensure the dead letter queue is not PUT disabled.
- Ensure the dead letter queue is not at the MAXMSG limit.

CSQQ113I: psb-name region-id This message cannot be processed

- Check the CSQOUTX data set to determine the cause of the CSQINPX failure.
- Some commands may not be processed.

CSQX035I: csect-name Connection to queue manager qmgr-name stopping or broken, MQCC= mqcc MQRC= mqrc (mqrc-text)

- Check the MQRC to determine the cause of the failure.
- These codes are documented in [IBM MQ for z/OS messages, completion, and reason codes](#).

CSQX032I: csect-name Initialization command handler terminated

- Check the MQRC to determine the cause of the failure.
- These codes are documented in [IBM MQ for z/OS messages, completion, and reason codes](#).

CSQX048I: csect-name Unable to convert message for name, MQCC= mqcc MQRC= mqrc (mqrc-text)

- Check the joblog to determine the cause of the TCP/IP failure.
- Check the TCP/IP address space for errors.

CSQX234I: csect-name Listener stopped, TRPTYPE= trptype INDISP= disposition

- If the listener does not stop, following a **STOP** command, check the TCP/IP address space for errors.
- Follow the system programmer response.

CSQX407I: csect-name Cluster queue q-name definitions inconsistent

- Multiple cluster queues within the cluster have inconsistent values. Investigate and resolve the differences.

CSQX411I: csect-name Repository manager stopped

- If the repository manager has stopped because of an error, check the joblog for messages.

CSQX417I: csect-name Cluster-senders remain for removed queue manager qmgr-name

- Follow the system programmer response.

CSQX418I: csect-name Only one repository for cluster cluster_name

- For increased high availability, clusters should be configured with two full repositories.

CSQX419I: csect-name No cluster-receivers for cluster cluster_name

- Follow the system programmer response.

CSQX420I: csect-name No repositories for cluster cluster_name

- Follow the system programmer response.

CSQX448E: csect-name Repository manager stopping because of errors. Restart in n seconds

- Follow the system programmer response.

This message is put out every 600 seconds (10 minutes) until the SYSTEM.CLUSTER.COMMAND.QUEUE is enabled, by using the command:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

Before enabling the queue, manual intervention might be required to resolve the problem that caused the repository manager to end, prior to the first CSQX448E message being issued.

CSQX548E: csect-name Messages sent to local dead-letter queue, channel channel-name reason=mqrc (mqrc-text)

- Follow the system programmer response.

CSQX788I: csect-name DNS lookup for address address using function 'func' took n seconds

- Follow the system programmer response.

CSQY225E: csect-name Queue manager is critically short of local storage above the bar - take action


- The queue manager is running critically short of virtual storage above the bar. Action should be taken to relieve the situation, and to avoid the possible abnormal termination of the queue manager.

CSQ5038I: csect-name Service task service-task has been unresponsive since hh.mm.ss.nnnnnn. Check for problems with Db2

- Follow the system programmer response.

Configurazioni HA (High Availability)

Se si desidera utilizzare i gestori code IBM MQ in una configurazione HA (High Availability), è possibile impostare i gestori code in modo che funzionino con un gestore HA (High Availability), come ad esempio PowerHA per AIX (in precedenza HACMP) o il servizio cluster Microsoft (MSCS) o con gestori code a più istanze IBM MQ. Su sistemi Linux, è anche possibile distribuire i gestori code di dati replicati (RDQM), che utilizzano un gruppo basato sul quorum per fornire alta disponibilità. Un'altra opzione, HA nativa, è destinata alle distribuzioni del contenitore.

 Un'altra opzione per una soluzione di ripristino di emergenza o alta disponibilità è quella di distribuire una coppia di dispositivi IBM MQ. Vedi [High Availability](#) e [Disaster Recovery](#) nella documentazione di IBM MQ Appliance.

È necessario essere consapevoli delle seguenti definizioni di configurazione:

Cluster gestore code

Gruppi di due o più gestori code su uno o più computer, che forniscono l'interconnessione automatica e che consentono la condivisione delle code per il bilanciamento del carico e la ridondanza. Da IBM WebSphere MQ 7.1 in poi, il ripristino da errori del cluster riesegui le operazioni che hanno causato problemi fino a quando i problemi non vengono risolti.

Cluster HA

I cluster HA sono gruppi di due o più computer e risorse, come dischi e reti, collegati e configurati in modo tale che, in caso di errore, un gestore HA (High Availability), come HACMP (AIX and Linux) o MSCS (Windows) esegua un *failover*. Il failover trasferisce i dati di stato delle applicazioni dal computer in errore ad un altro computer nel cluster e ne riavvia l'operazione. Ciò fornisce l'alta disponibilità dei servizi in esecuzione all'interno del cluster HA. La relazione tra cluster IBM MQ e cluster HA è descritta in [“Relazione tra cluster HA e cluster del gestore code”](#) a pagina 496.

Gestori code a più istanze

Istanze dello stesso gestore code configurate su due o più computer. Avviando più istanze, un'istanza diventa l'istanza attiva e le altre istanze diventano standby. Se l'istanza attiva ha esito negativo, un'istanza in standby in esecuzione su un computer differente prende automaticamente il sopravvento. È possibile utilizzare i gestori code a più istanze per configurare i propri sistemi di messaggistica altamente disponibili basati su IBM MQ, senza richiedere una tecnologia cluster come HACMP o MSCS. I cluster HA e i gestori code a più istanze sono modi alternativi per rendere i gestori code altamente disponibili. Non combinarli inserendo un gestore code a più istanze in un cluster HA.

Gestori code di dati replicati ad alta disponibilità (RDQM HA)

Istanze dello stesso gestore code configurate su ciascun nodo in un gruppo di tre server Linux. Una delle tre istanze è l'istanza attiva. I dati del gestore code attivo vengono replicati in modo sincrono nelle altre due istanze, quindi una di queste istanze può assumere il controllo in caso di errore. Il raggruppamento dei server è controllato da Pacemaker e la replica da DRBD.

Gestori code di dati replicati di ripristino di emergenza (RDQM DR)

Un gestore code viene eseguito su un nodo primario in un sito, con un'istanza secondaria di tale gestore code ubicata su un nodo di recupero in un sito differente. I dati vengono replicati tra l'istanza primaria e l'istanza secondaria, e se il nodo primario viene perso per qualche motivo, l'istanza secondaria può essere creata nell'istanza primaria e avviata. Entrambi i nodi devono essere server Linux. La replica è controllata da DRBD.

Gestori code di dati replicati di ripristino di emergenza / alta disponibilità (RDQM DR/HA)

È possibile configurare un gestore code di dati replicati (RDQM) che viene eseguito su un gruppo ad alta disponibilità su un sito, ma può eseguire il failover su un altro gruppo ad alta disponibilità su un altro sito se si verifica una situazione di emergenza che rende il primo gruppo non disponibile. Questo è noto come DR/HA RDQM.

CP4I HA nativa

La HA nativa è una soluzione di alta disponibilità destinata alle distribuzioni del contenitore di IBM MQ. La HA nativa utilizza la replica del log per mantenere aggiornate tre istanze di un gestore code in esecuzione su nodi differenti. Un'istanza è attiva in qualsiasi momento ed elabora i messaggi. Il gestore code attivo invia i propri aggiornamenti log alle altre due istanze per mantenerli aggiornati. Se l'istanza attiva ha esito negativo, una delle istanze di replica assume automaticamente il ruolo attivo.

Differenze tra gestori code a più istanze e cluster HA

I gestori code a più istanze e i cluster HA sono modi alternativi per ottenere l'elevata disponibilità per i gestori code. Ecco alcuni punti che evidenziano le differenze tra i due approcci.

I gestori code a più istanze includono le seguenti funzioni:

- Supporto failover di base integrato in IBM MQ
- Failover più rapido rispetto al cluster HA
- Configurazione e funzionamento semplici
- Integrazione con IBM MQ Explorer

Le limitazioni dei gestori code a più istanze includono:

- Storage di rete altamente disponibile e ad alte prestazioni richiesto
- Configurazione di rete più complessa perché il gestore code modifica l'indirizzo IP quando si verifica il failover

I cluster HA includono le seguenti funzioni:

- La capacità di coordinare più risorse, come un server delle applicazioni o un database
- Opzioni di configurazione più flessibili, inclusi i cluster che comprendono più di due nodi
- È possibile eseguire il failover più volte senza l'intervento dell'operatore
- Acquisizione dell'indirizzo IP del gestore code come parte del failover

Le limitazioni dei cluster HA includono:

- Sono richiesti ulteriori acquisti di prodotti e competenze
- I dischi che possono essere commutati tra i nodi del cluster sono obbligatori
- La configurazione dei cluster HA è relativamente complessa
- Il failover è piuttosto lento storicamente, ma i prodotti cluster HA recenti stanno migliorando
- I failover non necessari possono verificarsi se si verificano delle carenze negli script utilizzati per monitorare le risorse, ad esempio i gestori code

Relazione tra cluster HA e cluster del gestore code

I cluster del gestore code forniscono il bilanciamento del carico dei messaggi tra le istanze disponibili delle code del gestore code. Ciò offre una disponibilità superiore rispetto a un singolo gestore code poiché, in caso di errore di un gestore code, le applicazioni di messaggistica possono ancora inviare

messaggi e accedere alle istanze rimanenti di una coda del cluster del gestore code. Tuttavia, sebbene i cluster dei gestori code instradino automaticamente i nuovi messaggi ai gestori code disponibili in un cluster, i messaggi attualmente accodati su un gestore code non disponibile non saranno disponibili fino a quando tale gestore code non viene riavviato. Per questo motivo, i cluster di gestori code da soli non forniscono l'alta disponibilità di tutti i dati dei messaggi o forniscono il rilevamento automatico dell'errore del gestore code e l'attivazione automatica del riavvio o del failover del gestore code. I cluster HA (High Availability) forniscono queste funzioni. I due tipi di cluster possono essere utilizzati insieme per un buon effetto. Per un'introduzione ai cluster dei gestori code, consultare [Progettazione dei cluster](#).

Concetti correlati

[MQ Adv.](#) [Linux](#) [CD](#) [Alta disponibilità per IBM MQ Advanced container](#)

[Linux](#) [AIX](#) **Cluster HA su AIX and Linux**

È possibile utilizzare IBM MQ con un cluster HA (high availability) su piattaforme AIX and Linux : ad esempio, PowerHA per AIX (in precedenza HACMP), Veritas Cluster Server, HP Serviceguard o un cluster Red Hat Enterprise Linux con Red Hat Cluster Suite.

Questa sezione introduce [“Configurazioni cluster HA” a pagina 497](#), la relazione tra i cluster HA e i cluster dei gestori code, [“IBM MQ client” a pagina 498](#)e [“IBM MQ che opera in un cluster HA” a pagina 498](#)e guida l'utente attraverso i passi e fornisce script di esempio che è possibile adattare per configurare i gestori code con un cluster HA.

Fare riferimento alla documentazione del cluster HA specifica per il proprio ambiente per assistenza con i passi di configurazione descritti in questa sezione.

Configurazioni cluster HA

In questa sezione, il termine *nodo* viene utilizzato per fare riferimento all'entità che sta eseguendo un sistema operativo e il software HA; "computer", "sistema" o "macchina" o "partizione" o "blade" potrebbero essere considerati sinonimi in questo utilizzo. È possibile utilizzare IBM MQ per configurare le configurazioni di standby o di takeover, incluso il takeover reciproco in cui tutti i nodi cluster eseguono il carico di lavoro IBM MQ .

Una configurazione *standby* è la configurazione cluster HA di base in cui un nodo esegue il lavoro mentre l'altro nodo agisce solo come standby. Il nodo standby non esegue il lavoro ed è indicato come inattivo; questa configurazione è talvolta denominata *standby a freddo*. Tale configurazione richiede un alto grado di ridondanza hardware. Per risparmiare sull'hardware, è possibile estendere questa configurazione per avere più nodi di lavoro con un singolo nodo di standby. Il punto è che il nodo di standby può assumere il controllo del lavoro di qualsiasi altro nodo di lavoro. Questa configurazione è ancora indicata come configurazione standby e a volte come configurazione "N+1".

Una configurazione di *takeover* è una configurazione più avanzata in cui tutti i nodi eseguono del lavoro e il lavoro critico può essere assunto in caso di un errore del nodo.

Una configurazione di *takeover unilaterale* è una configurazione in cui un nodo standby esegue del lavoro aggiuntivo, non critico e non rimovibile. Questa configurazione è simile a una configurazione standby, ma con un lavoro (non critico) eseguito dal nodo standby.

Una configurazione di *takeover reciproco* è una configurazione in cui tutti i nodi eseguono operazioni ad alta disponibilità (mobili). Questo tipo di configurazione del cluster HA è anche a volte indicato come "Attivo / Attivo" per indicare che tutti i nodi stanno elaborando attivamente il carico di lavoro critico.

Con la configurazione di standby estesa o con una delle configurazioni di takeover, è importante considerare il carico di picco che potrebbe essere posizionato su un nodo che può assumere il controllo del lavoro di altri nodi. Tale nodo deve disporre di una capacità sufficiente per mantenere un livello di prestazioni accettabile.

Relazione tra cluster HA e cluster del gestore code

I cluster del gestore code riducono l'amministrazione e forniscono il bilanciamento del carico dei messaggi tra le istanze delle code del cluster del gestore code. Inoltre, offrono una disponibilità superiore rispetto a un singolo gestore code poiché, in seguito a un malfunzionamento di un gestore code, le applicazioni di messaggistica possono ancora accedere alle istanze rimanenti di una coda cluster del gestore code. Tuttavia, i cluster del gestore code da soli non forniscono il rilevamento automatico dell'errore del gestore code e l'attivazione automatica del riavvio o del failover del gestore code. I cluster HA forniscono queste funzioni. I due tipi di cluster possono essere utilizzati insieme per un buon effetto.

IBM MQ client

I client IBM MQ che stanno comunicando con un gestore code che potrebbe essere soggetto a un riavvio o a un takeover devono essere scritti per tollerare una connessione interrotta e devono tentare ripetutamente di riconnettersi. IBM MQ include funzioni nell'elaborazione di CCDT (Client Channel Definition Table) che aiutano con la disponibilità della connessione e il bilanciamento del carico di lavoro; tuttavia, queste non sono direttamente rilevanti quando si utilizza un sistema di failover.

La funzionalità transazionale consente a IBM MQ MQI client di partecipare a transazioni a due fasi, purché il client sia connesso allo stesso gestore code. La funzionalità transazionale non può utilizzare tecniche, come un programma di bilanciamento del carico IP, per effettuare una selezione da un elenco di gestori code. Quando si utilizza un prodotto HA, un gestore code conserva la propria identità (nome e indirizzo) indipendentemente dal nodo su cui è in esecuzione, quindi è possibile utilizzare la funzionalità transazionale con i gestori code che si trovano sotto il controllo HA.

IBM MQ che opera in un cluster HA

Tutti i cluster HA hanno il concetto di un'unità di failover. Questa è una serie di definizioni che contiene tutte le risorse che costituiscono il servizio ad elevata disponibilità. L'unità di failover comprende il servizio stesso e tutte le altre risorse da cui dipende.

Le soluzioni HA utilizzano termini differenti per un'unità di failover:

- Su PowerHA per AIX l'unità di failover è denominata *gruppo di risorse*.
- Su Veritas Cluster Server è noto come *gruppo di servizio*.
- Su Serviceguard viene chiamato *pacchetto*.

Questo argomento usa il termine *gruppo di risorse* per indicare un'unità di failover.

L'unità di failover più piccola per IBM MQ è un gestore code. Generalmente, il gruppo di risorse che contiene il gestore code contiene anche dischi condivisi in un gruppo di volumi o in un gruppo di dischi riservato esclusivamente per l'utilizzo da parte del gruppo di risorse e l'indirizzo IP utilizzato per la connessione al gestore code. È anche possibile includere altre risorse IBM MQ, come un listener o un controllo trigger nello stesso gruppo di risorse, come risorse separate o sotto il controllo del gestore code stesso.

Un gestore code che deve essere utilizzato in un cluster HA deve avere i propri dati e log sui dischi condivisi tra i nodi nel cluster. Il cluster HA garantisce che solo un nodo alla volta possa scrivere sui dischi. Il cluster HA può utilizzare uno script di controllo per monitorare lo stato del gestore code.

È possibile utilizzare un singolo disco condiviso sia per i dati che per i log correlati al gestore code. Tuttavia, è normale utilizzare file system condivisi separati in modo che possano essere ridimensionati e ottimizzati in modo indipendente.

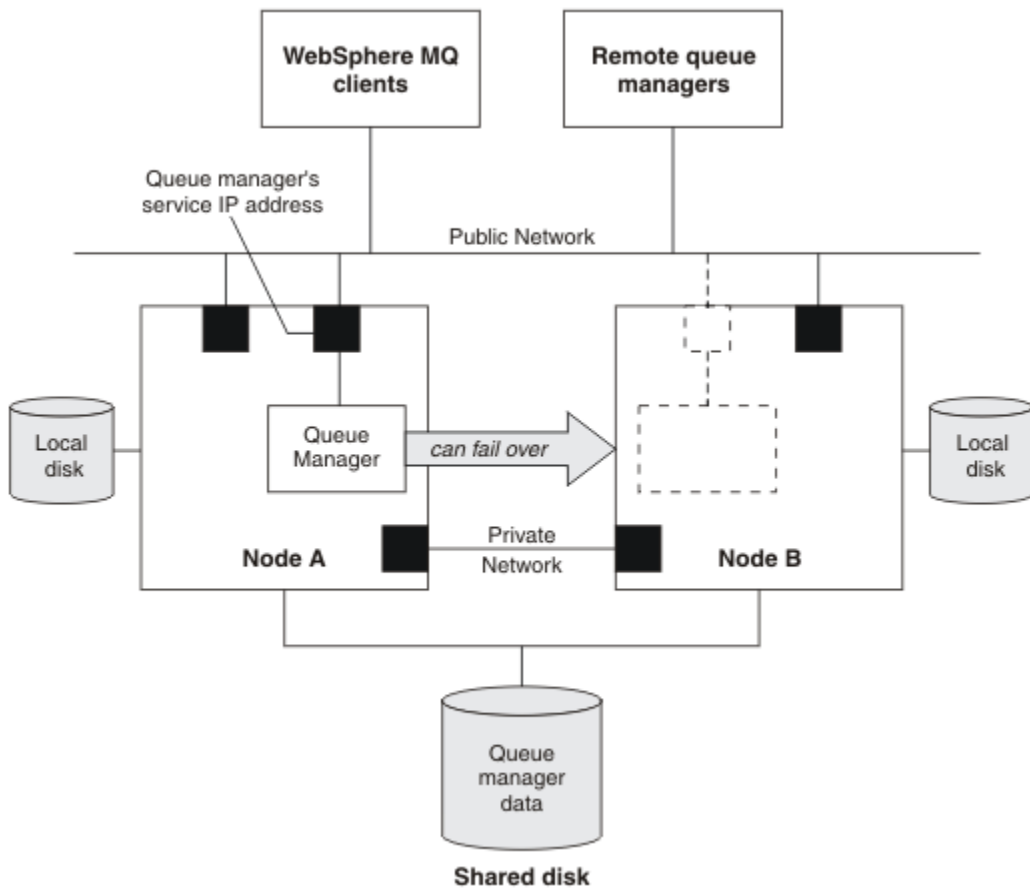


Figura 69. Cluster HA

La Figura 1 illustra un cluster HA con due nodi. Il cluster HA sta gestendo la disponibilità di un gestore code che è stato definito in un gruppo di risorse. Si tratta di una configurazione attiva / passiva o di standby a freddo, poiché solo un nodo, il nodo A, sta attualmente eseguendo un gestore code. Il gestore code è stato creato con dati e file di log su un disco condiviso. Il gestore code ha un indirizzo IP del servizio che è anche gestito dal cluster HA. Il gestore code dipende dal disco condiviso e dall'indirizzo IP del servizio. Quando il cluster HA riporta un errore sul gestore code dal nodo A al nodo B, sposta prima le risorse dipendenti del gestore code sul nodo B, quindi avvia il gestore code.

Se il cluster HA contiene più di un gestore code, la configurazione del cluster HA potrebbe comportare l'esecuzione di due o più gestori code sullo stesso nodo dopo un failover. A ogni gestore code nel cluster HA deve essere assegnato il suo proprio numero di porta, che utilizza su qualsiasi nodo cluster che sia attivo in un determinato momento.

Generalmente, il cluster HA viene eseguito come utente root. IBM MQ viene eseguito come utente mqm. L'amministrazione di IBM MQ è concessa a membri del gruppo mqm. Verificare che l'utente e il gruppo mqm esistano entrambi su tutti i nodi cluster HA. L'ID utente e l'ID gruppo devono essere congruenti nel cluster. La gestione di IBM MQ da parte dell'utente root non è consentita; gli script di avvio, arresto o monitoraggio devono passare all'utente mqm.

Nota: IBM MQ deve essere installato correttamente su tutti i nodi; non è possibile condividere i file eseguibili del prodotto.

Linux AIX **Configurazione dei dischi condivisi su AIX and Linux**

Un gestore code IBM MQ in un cluster HA richiede che i file di dati e i file di log si trovino in file system remoti denominati comuni su un disco condiviso.

Informazioni su questa attività

La Figura 1 mostra un layout possibile per un gestore code in un cluster HA. I dati del gestore code e le directory di log si trovano entrambi sul disco condiviso montato su /MQHA/QM1. Questo disco viene commutato tra i nodi del cluster HA quando si verifica il failover in modo che i dati siano disponibili ovunque venga riavviato il gestore code. Il file mqs.ini ha una stanza per il gestore code QM1. La stanza Log nel file qm.ini ha un valore per LogPath.

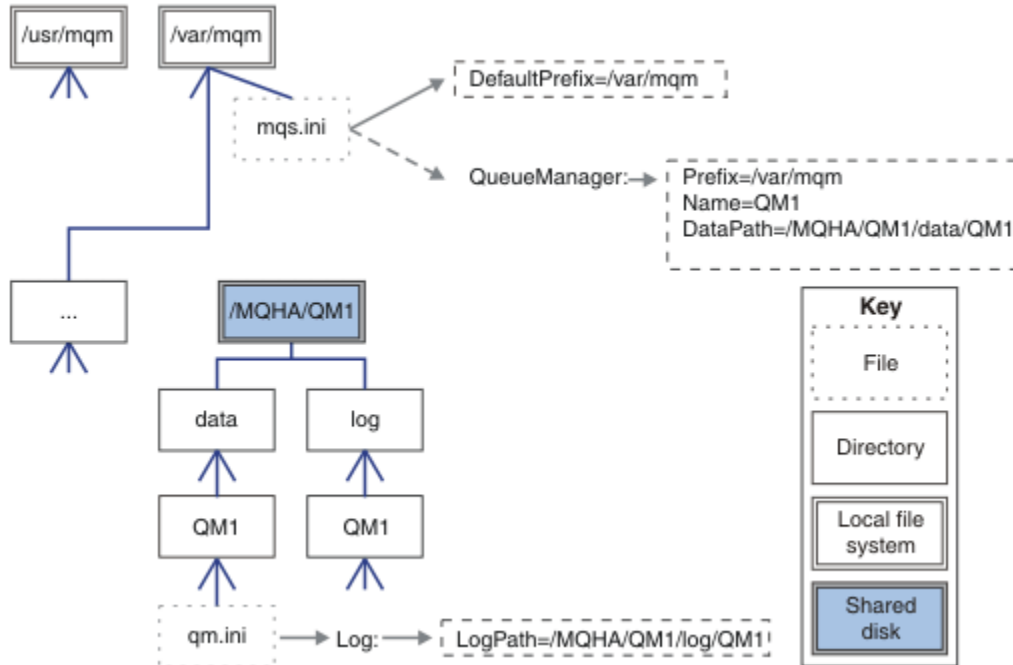


Figura 70. Directory data e log denominate condivise

Procedura

- Decidere i nomi dei punti di montaggio per i filesystem del gestore code.
Ad esempio, /MQHA/qmgrname/data per i file di dati del gestore code e /MQHA/qmgrname/log per i relativi file di log.
- Creare un gruppo di volumi (o un gruppo di dischi) per contenere i dati del gestore code e i file di log.
Questo gruppo di volume è gestito da un cluster HA (High Availability) nello stesso gruppo di risorse del gestore code.
- Creare i file system per i dati del gestore code e i file di log nel gruppo volumi.
- Per ogni nodo a turno, creare i punti di montaggio per i filesystem e assicurarsi che i filesystem possano essere montati.
L'utente mqm deve possedere i punti di montaggio.

Linux > AIX Creazione di un gestore code cluster HA su AIX and Linux

Il primo passo per utilizzare un gestore code in un cluster ad alta disponibilità consiste nel creare il gestore code su uno dei nodi.

Informazioni su questa attività

Per creare un gestore code da utilizzare in un cluster HA, è necessario selezionare prima uno dei nodi nel cluster su cui creare il gestore code, quindi completare la seguente procedura su questo nodo.

Procedura

- Montare i file system del gestore code sul nodo.

2. Creare il gestore code utilizzando il comando **crtmqm**.

Ad esempio:

```
crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
```

3. Avviare manualmente il gestore code utilizzando il comando **strmqm**.
4. Completare qualsiasi configurazione iniziale del gestore code, ad esempio la creazione di code e canali e l'impostazione del gestore code per avviare automaticamente un listener all'avvio del gestore code.
5. Arrestare il gestore code utilizzando il comando **endmqm**.
6. Utilizzare il comando **dspmqlnf** per visualizzare il comando **addmqinf**:

```
dspmqlnf -o command qmgrname
```

dove qmgrname è il nome del gestore code.

Per ulteriori informazioni sull'utilizzo del comando **addmqinf**, consultare [“Aggiunta della configurazione del gestore code ad altri nodi cluster HA su AIX and Linux” a pagina 501.](#)

Il comando **addmqinf** viene visualizzato in modo simile al seguente esempio:

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

7. Prendere nota del comando visualizzato.
8. Smontare i filesystem del gestore code.

Operazioni successive

Si è ora pronti a completare la procedura descritta in [“Aggiunta della configurazione del gestore code ad altri nodi cluster HA su AIX and Linux” a pagina 501.](#)

Aggiunta della configurazione del gestore code ad altri nodi cluster HA su AIX and Linux

È necessario aggiungere le informazioni di configurazione del gestore code agli altri nodi nel cluster HA.

Prima di iniziare

Prima di completare questa attività, è necessario aver completato i passi in [“Creazione di un gestore code cluster HA su AIX and Linux” a pagina 500.](#) Una volta creato il gestore code, è necessario aggiungere le informazioni di configurazione per il gestore code a ciascuno degli altri nodi nel cluster HA completando la seguente procedura su ognuno degli altri nodi.

Informazioni su questa attività

Quando si crea un gestore code da utilizzare in un cluster HA, è necessario prima selezionare uno dei nodi nel cluster su cui creare il gestore code, come descritto in [“Creazione di un gestore code cluster HA su AIX and Linux” a pagina 500.](#)

Procedura

1. Montare i filesystem del gestore code.
2. Aggiungere le informazioni di configurazione del gestore code al nodo.
Esistono due metodi per aggiungere le informazioni di configurazione:
 - Modificando direttamente `/var/mqm/mqs.ini`.
 - Immettendo il comando **addmqinf** visualizzato dal comando **dspmqlnf** nel passo 6 in [“Creazione di un gestore code cluster HA su AIX and Linux” a pagina 500.](#)
3. Avviare e arrestare il gestore code per verificare la configurazione.

I comandi utilizzati per avviare e arrestare il gestore code devono essere emessi dalla stessa installazione IBM MQ del comando **addmqinf**. Per avviare e arrestare il gestore code da un'installazione diversa da quella attualmente associata al gestore code, è necessario prima impostare l'installazione associata al gestore code utilizzando il comando **setmqm**. Per ulteriori informazioni, vedere **setmqm**.

4. Smontare i filesystem del gestore code.

Linux AIX **Script shell di esempio per l'avvio di un gestore code cluster HA su AIX and Linux**

Il gestore code è rappresentato nel cluster HA come risorsa. Il cluster HA deve essere in grado di avviare e arrestare il gestore code. Nella maggior parte dei casi, è possibile utilizzare uno script shell per avviare il gestore code. È necessario rendere questi script disponibili nella stessa ubicazione su tutti i nodi nel cluster, utilizzando un file system di rete o copiandoli su ciascuno dei dischi locali.

Nota: Prima di riavviare un gestore code non riuscito, è necessario disconnettere le applicazioni da tale istanza del gestore code. In caso contrario, il gestore code potrebbe non essere riavviato correttamente.

Di seguito sono riportati esempi di script shell adatti. Puoi personalizzarli in base alle tue esigenze e utilizzarli per avviare il gestore code sotto il controllo del tuo cluster HA.

Il seguente script di shell è un esempio di come passare dall'utente cluster HA all'utente mqm in modo che il gestore code possa essere correttamente avviato:

```
#!/bin/ksh
# A simple wrapper script to switch to the mqm user.
su mqm -c name_of_your_script $*
```

Il seguente script di shell è un esempio di come avviare un gestore code senza fare supposizioni sullo stato corrente del gestore code. Si noti che utilizza un metodo estremamente brusco per terminare i processi che appartengono al gestore code:

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.

# The only argument is the queue manager name. Save it as QM variable
QM=$1

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

# End any queue manager processes which might be running.
srchstr="(|-m)$QM *.*$"
for process in amqzmuc0 amqzxma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrista amqirmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
do
  ps -ef | tr "\t" " " | grep $process | grep -v grep | \
  egrep "$srchstr" | awk '{print $2}' | \
  xargs kill -9 > /dev/null 2>&1
done

# It is now safe to start the queue manager.
# The strmqm command does not use the -x flag.
strmqm ${QM}
```

È possibile modificare lo script per avviare altri programmi correlati.

cluster HA su AIX and Linux

Nella maggior parte dei casi, è possibile utilizzare uno script di shell per arrestare un gestore code. Di seguito sono riportati esempi di script shell adatti. Puoi personalizzarli in base alle tue esigenze e utilizzarli per arrestare il gestore code sotto il controllo del tuo cluster HA.

Il seguente script è un esempio di come arrestare immediatamente un gestore code senza fare ipotesi sullo stato corrente del gestore code. Lo script deve essere eseguito dall'utente mqm. Potrebbe quindi essere necessario racchiudere questo script in uno script shell per passare l'utente dall'utente del cluster HA a mqm. (Uno script shell di esempio è fornito in ["Script shell di esempio per l'avvio di un gestore code cluster HA su AIX and Linux"](#) a pagina 502.)

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi

if [ -z "$TIMEOUT" ]
then
    echo "ERROR! No timeout specified"
    exit 1
fi

for severity in immediate brutal
do
    # End the queue manager in the background to avoid
    # it blocking indefinitely. Run the TIMEOUT timer
    # at the same time to interrupt the attempt, and try a
    # more forceful version. If the brutal version fails,
    # nothing more can be done here.

    echo "Attempting ${severity} end of queue manager '${QM}'"
    case $severity in

    immediate)
        # Minimum severity of endmqm is immediate which severs connections.
        # HA cluster should not be delayed by clients
        endmqm -i ${QM} &
        ;;

    brutal)
        # This is a forced means of stopping queue manager processes.

        srchstr="( |-m)$QM *.*$"
        for process in amqzmuc0 amqzxa0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
            amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqrmppa \
            amqzfuma amqzmuf0 amqzmur0 amqzmgr0

        do
            ps -ef | tr "\t" " " | grep $process | grep -v grep | \
                egrep "$srchstr" | awk '{print $2}' | \
                xargs kill -9 > /dev/null 2>&1
        done

    esac

    TIMED_OUT=yes
    SECONDS=0
    while (( $SECONDS < ${TIMEOUT} ))
    do
        TIMED_OUT=yes
        i=0
        while [ $i -lt 5 ]
```

```

do
  # Check for execution controller termination
  srchstr="( |-m)$QM *.*$"
  cnt=`ps -ef | tr "\t" " " | grep amqzma0 | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}' | wc -l`
  i=`expr $i + 1`
  sleep 1
  if [ $cnt -eq 0 ]
  then
    TIMED_OUT=no
    break
  fi
done

if [ ${TIMED_OUT} = "no" ]
then
  break
fi

echo "Waiting for ${severity} end of queue manager '${QM}'"
sleep 1
done # timeout loop

if [ ${TIMED_OUT} = "yes" ]
then
  continue      # to next level of urgency
else
  break         # queue manager is ended, job is done
fi

done # next phase

```

Nota: In base ai processi in esecuzione per un gestore code specifico, l'elenco dei processi del gestore code inclusi in questo script potrebbe non essere completo o potrebbe includere più processi rispetto ai processi in esecuzione per tale gestore code:

```

for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
  amqzlsa0 runmqchi runmqlsr amqcrista amqirmfa amqimppa \
  amqzfuma amqmuf0 amqzmur0 amqzmgr0

```

Un processo può essere incluso o escluso da questo elenco in base alla funzione configurata e ai processi in esecuzione per un gestore code specifico. Per un elenco completo di processi e informazioni sull'arresto dei processi in un ordine specifico, consultare [Arresto manuale di un gestore code in UNIX e Linux](#).

Linux → AIX **Controllo di un gestore code cluster HA su AIX and Linux**

Generalmente, il cluster HA (High Availability) può monitorare periodicamente lo stato del gestore code. Nella maggior parte dei casi, è possibile utilizzare uno script di shell per questo. Di seguito sono riportati esempi di script shell adatti. È possibile personalizzare questi script in base alle proprie esigenze e utilizzarli per effettuare ulteriori controlli di monitoraggio specifici per il proprio ambiente.

È possibile che più installazioni di IBM MQ coesistano su un sistema. Per ulteriori informazioni su più installazioni, consultare [Più installazioni](#). Se si intende utilizzare lo script di monitoraggio su più installazioni, potrebbe essere necessario eseguire alcune operazioni aggiuntive. Se si dispone di un'installazione primaria, non è necessario specificare `MQ_INSTALLATION_PATH` per utilizzare lo script. In caso contrario, utilizzare la seguente procedura per accertarsi che `MQ_INSTALLATION_PATH` sia identificato correttamente:

1. Utilizzare il comando **crtmqenv** da un'installazione IBM MQ per identificare il `MQ_INSTALLATION_PATH` corretto per un gestore code:

```
crtmqenv -m qmname
```

Questo comando restituisce il valore `MQ_INSTALLATION_PATH` corretto per il gestore code specificato da `qmname`.

2. Eseguire lo script di monitoraggio con i parametri `qmname` e `MQ_INSTALLATION_PATH` appropriati.

Nota: PowerHA per AIX non fornisce un modo per fornire un parametro al programma di controllo per il gestore code. È necessario creare un programma di monitoraggio separato per ogni gestore code, che incapsula il nome del gestore code. Di seguito è riportato un esempio di script utilizzato su AIX per incapsulare il nome del gestore code:

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname MQ_INSTALLATION_PATH
```

dove `MQ_INSTALLATION_PATH` è un parametro facoltativo che specifica il percorso di installazione di IBM MQ a cui è associato il gestore code `qmname`.

Il seguente script non è valido per la possibilità che **runmqsc** si blocchi. In genere, i cluster HA trattano uno script di monitoraggio in sospeso come un errore e sono essi stessi robusti a questa possibilità.

Tuttavia, lo script tollera che il gestore code si trovi nello stato di avvio. Ciò è dovuto al fatto che è comune per il cluster HA avviare il monitoraggio del gestore code non appena viene avviato. Alcuni cluster HA distinguono tra una fase di avvio e una fase di esecuzione per le risorse, ma è necessario configurare la durata della fase di avvio. Poiché il tempo impiegato per avviare un gestore code dipende dalla quantità di lavoro che deve eseguire, è difficile scegliere il tempo massimo impiegato per avviare un gestore code. Se si sceglie un valore troppo basso, il cluster HA assume erroneamente che il gestore code abbia avuto esito negativo quando non è stato completato l'avvio. Ciò potrebbe causare una sequenza infinita di failover.

Questo script deve essere eseguito dall'utente mqm; potrebbe quindi essere necessario racchiudere questo script in uno script shell per passare l'utente dall'utente del cluster HA a mqm (uno script shell di esempio viene fornito in [“Script shell di esempio per l'avvio di un gestore code cluster HA su AIX and Linux” a pagina 502](#)):

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0 => Either the queue manager is starting or the queue manager is running and responds.
#     Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
    # No path specified, assume system primary install or MQ level < 7.1.0.0
    echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
    echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
    PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

    echo "Queue manager '${QM}' is responsive"
    result=0

else # ping failed

    # Don't condemn the queue manager immediately, it might be starting.
    srchstr="( |m)$QM *.*$"
    cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
        | awk '{print $2}' | wc -l`
    if [ $cnt -gt 0 ]
```

```

then
# It appears that the queue manager is still starting up, tolerate
echo "Queue manager '${QM}' is starting"
result=0
else
# There is no sign of the queue manager starting
echo "Queue manager '${QM}' is not responsive"
result=$pingresult
fi

fi

exit $result

```

Linux

AIX

Inserimento del gestore code sotto il controllo del cluster HA su

AIX and Linux

È necessario configurare il gestore code, sotto il controllo del cluster HA, con l'indirizzo IP del gestore code e i dischi condivisi.

Informazioni su questa attività

Per mettere il gestore code sotto il controllo di un cluster HA, è necessario definire un gruppo di risorse che contenga il gestore code e tutte le risorse associate.

Procedura

1. Creare il gruppo di risorse contenente il gestore code, il volume del gestore code o il gruppo di dischi e l'indirizzo IP del gestore code.
L'indirizzo IP è un indirizzo IP virtuale, non l'indirizzo IP del computer.
2. Verificare che il cluster HA commuta correttamente le risorse tra i nodi del cluster e che sia pronto a controllare il gestore code.

Linux

AIX

Eliminazione di un gestore code del cluster HA su AIX and Linux

Si potrebbe voler rimuovere un gestore code da un nodo che non è più richiesto per eseguire il gestore code.

Informazioni su questa attività

Per rimuovere il gestore code da un nodo in un cluster HA, è necessario rimuovere le informazioni di configurazione.

Procedura

1. Rimuovere il nodo dal cluster HA in modo che il cluster HA non tenterà più di attivare il gestore code su questo nodo.
2. Utilizzare il seguente comando **rmvmqinf** per eliminare le informazioni di configurazione del gestore code:

`rmvmqinf qmgrname`
3. Opzionale: Per eliminare completamente il gestore code, utilizzare il comando **dltmqm**.

Importante: Tenere presente che l'eliminazione del gestore code utilizzando il comando **dltmqm** elimina completamente i dati e i file di log del gestore code.

Una volta eliminato il gestore code, è possibile utilizzare il comando **rmvmqinf** per rimuovere le restanti informazioni di configurazione dagli altri nodi.

Windows

Supporto Microsoft Cluster Service (MSCS)

Introduzione e configurazione di MSCS per supportare il failover di server virtuali. MSCS è noto anche come Windows Server Failover Clustering (WSFC).

Queste informazioni si applicano solo a IBM MQ for Windows .

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

MSCS/WSFC consente di collegare i server in un cluster, offrendo una maggiore disponibilità di dati e applicazioni e semplificando la gestione del sistema. MSCS/WSFC è in grado di rilevare e ripristinare automaticamente gli errori del server o dell'applicazione.

MSCS/WSFC supporta il failover di server virtuali, che corrispondono ad applicazioni, siti web, code di stampa o condivisioni di file (inclusi, ad esempio, i relativi mandrini del disco, i file e gli indirizzi IP).

Failover è il processo mediante il quale MSCS/WSFC rileva un malfunzionamento in una applicazione su un computer nel cluster e arresta l'applicazione interrotta in modo ordinato, trasferisce i suoi dati di stato all'altro computer e reinizializza l'applicazione.

Per informazioni su come configurare e utilizzare i cluster di failover, consultare gli argomenti secondari.

Windows Introduzione ai cluster MSCS

I cluster Microsoft Cluster Service (MSCS) sono gruppi di due o più computer, connessi 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 iniziando nuovamente l'operazione.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

“Configurazioni HA (High Availability)” a pagina 495 contiene un confronto tra cluster MSCS, gestori code a più istanze e cluster IBM MQ .

In questa sezione e nei relativi argomenti subordinati, il termine *cluster*, se utilizzato da solo, **sempre** indica un cluster MSCS. Questo è diverso da un cluster IBM MQ descritto altrove in questa guida.

Un cluster a due macchine comprende due computer (ad esempio, A e B) che sono collegati insieme ad una rete per l'accesso client utilizzando un *indirizzo IP virtuale*. Possono anche essere connessi tra loro da una o più reti private. A e B condividono almeno un disco per le applicazioni server da utilizzare. C'è anche un altro disco condiviso, che deve essere un array ridondante di dischi indipendenti (*RAID*) Livello 1, per l'uso esclusivo di MSCS; è noto come disco *quorum* . MSCS controlla entrambi i computer per verificare che l'hardware e il software siano in esecuzione correttamente.

In una configurazione semplice come questa, entrambi i computer dispongono di tutte le applicazioni installate, ma solo il computer A viene eseguito con applicazioni attive; il computer B è solo in esecuzione e in attesa. Se il computer A rileva uno qualsiasi dei problemi, MSCS arresta l'applicazione interrotta in modo ordinato, trasferisce i suoi dati di stato all'altro computer e reinizializza l'applicazione. Questo è noto come *failover*. Le applicazioni possono essere rese *cluster - aware* in modo da interagire completamente con MSCS e failover.

Una configurazione tipica per un cluster a due computer è quella mostrata in [Figura 71 a pagina 508](#).

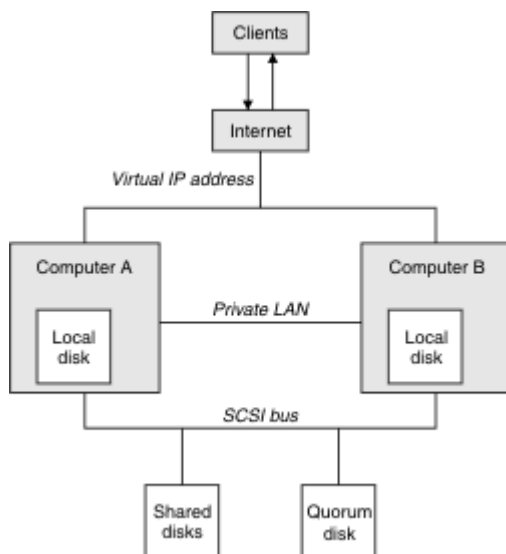


Figura 71. cluster MSCS a due computer

Ogni computer può accedere al disco condiviso, ma solo uno alla volta, sotto il controllo di MSCS. In caso di failover, MSCS cambia l'accesso all'altro computer. Il disco condiviso è di solito un RAID, ma non è necessario.

Ogni computer è collegato alla rete esterna per l'accesso client, e ognuno ha un indirizzo IP. Tuttavia, un client esterno, che comunica con questo cluster, è a conoscenza di un solo *indirizzo IP virtuale* e MSCS instrada il traffico IP all'interno del cluster in modo appropriato.

MSCS esegue anche le proprie comunicazioni tra i due computer, su una o più connessioni private o sulla rete pubblica, ad esempio per monitorare i loro stati utilizzando l'heartbeat e per sincronizzarne i database.

Windows **Impostazione di IBM MQ per il clustering MSCS**

Configurare IBM MQ per il cluster rendendo il gestore code l'unità di failover in MSCS. Definire un gestore code come una risorsa per MSCS, che può quindi monitorarlo e trasferirlo su un altro computer nel cluster se si verifica un problema.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Per impostare il sistema, si inizia installando IBM MQ su ogni computer del cluster.

Poiché il gestore code è associato con il nome di installazione IBM MQ, il nome di installazione di IBM MQ su tutti i computer nel cluster deve essere lo stesso. Consultare [Installazione e disinstallazione](#).

I gestori code stessi devono esistere solo sul computer su cui vengono creati. In caso di failover, MSCS avvia i gestori code sull'altro computer. I gestori code, tuttavia, devono avere i propri file di log e di dati su un disco condiviso del cluster e non su un'unità locale. Se si dispone già di un gestore code installato su un'unità locale, è possibile migrarlo utilizzando uno strumento fornito con IBM MQ; consultare [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511. Se si desidera creare nuovi gestori code da utilizzare con MSCS, consultare [“Creazione di un gestore code da utilizzare con MSCS”](#) a pagina 510.

Dopo l'installazione e la migrazione, utilizzare MSCS Cluster Administrator per rendere MSCS consapevole dei gestori code; consultare [“Inserimento di un gestore code sotto il controllo di MSCS”](#) a pagina 512.

Se si decide di rimuovere un gestore code dal controllo MSCS, utilizzare la procedura descritta in [“Rimozione di un gestore code dal controllo MSCS”](#) a pagina 519.

Windows *Imposta simmetria e MSCS*

Quando un'applicazione passa da un nodo all'altro, deve comportarsi nello stesso modo, indipendentemente dal nodo. Il modo migliore per garantire ciò è rendere gli ambienti identici.

Se è possibile, impostare un cluster con hardware, software del sistema operativo, software del prodotto e configurazione identici su ciascun computer. In particolare, assicurarsi che tutto il software richiesto installato sui due elaboratori sia identico in termini di versione, livello di manutenzione, SupportPacs, percorsi ed uscite e che sia presente uno spazio dei nomi comune (ambiente di sicurezza) come descritto in [“Sicurezza MSCS” a pagina 509](#).

Windows *Sicurezza MSCS*

Per una corretta sicurezza MSCS, seguire queste linee guida.

Le linee guida sono le seguenti:

- Accertarsi di disporre di installazioni software identiche su ciascun computer del cluster.
- Crea uno spazio dei nomi comune (ambiente di sicurezza) nel cluster.
- Creare i nodi dei membri del cluster MSCS di un dominio, all'interno dei quali l'account utente *proprietario cluster* è un account dominio.
- Creare gli altri account utente sul cluster anche gli account di dominio, in modo che siano disponibili su entrambi i nodi. Questo è automaticamente il caso se hai già un dominio e gli account rilevanti per IBM MQ sono account di dominio. Se al momento non si dispone di un dominio, prendere in considerazione la configurazione di un *mini - dominio* per soddisfare i nodi cluster e gli account pertinenti. Il vostro scopo è quello di rendere il vostro cluster di due computer come una singola risorsa di calcolo.

Tenere presente che un account locale su un computer non esiste sull'altro. Anche se si crea un account con lo stesso nome sull'altro computer, il relativo SID (security identifier) è diverso, in modo che, quando l'applicazione viene spostata sull'altro nodo, le autorizzazioni non esistono su tale nodo.

Durante un failover o uno spostamento, il supporto IBM MQ MSCS garantisce che tutti i file che contengono gli oggetti del gestore code abbiano autorizzazioni equivalenti sul nodo di destinazione. In modo esplicito, il codice verifica che gli amministratori e i gruppi mqm e l'account SYSTEM abbiano il controllo completo e che, se Everyone disponeva dell'accesso in lettura sul vecchio nodo, tale autorizzazione venga aggiunta sul nodo di destinazione.

Puoi utilizzare un account di dominio per eseguire il servizio IBM MQ . Assicurarsi che esista nel gruppo mqm locale su ciascun computer nel cluster.

Windows *Utilizzo di più gestori code con MSCS*

Se si sta eseguendo più di un gestore code su un computer, è possibile scegliere una di queste configurazioni.

Le configurazioni sono le seguenti:

- Tutti i gestori code in un singolo gruppo. In questa configurazione, se si verifica un problema con un qualsiasi gestore code, tutti i gestori code nel gruppo eseguono il failover sull'altro computer come un gruppo.
- Un singolo gestore code in ogni gruppo. In questa configurazione, se si verifica un problema con il gestore code, da solo esegue il failover sull'altro computer senza influire sugli altri gestori code.
- Una miscela delle prime due configurazioni.

Windows *Modalità cluster e MSCS*

Esistono due modalità in cui è possibile eseguire un sistema cluster con IBM MQ su Windows: Attivo / Passivo o Attivo / Attivo.

Nota: Se si utilizza MSCS insieme a COM + (Microsoft Transaction Server), non è possibile utilizzare la modalità Attiva / Attiva.

Modalità attiva / passiva

In modalità Attiva / Passiva, il computer A ha l'applicazione in esecuzione su di esso e il computer B è il backup, utilizzato solo quando MSCS rileva un problema.

È possibile utilizzare questa modalità con un solo disco condiviso, ma, se un'applicazione causa un failover, **tutte** le applicazioni devono essere trasferite come un gruppo (poiché solo un computer può accedere al disco condiviso alla volta).

È possibile configurare MSCS con A come computer *preferito*. Quindi, quando il computer A è stato riparato o sostituito e funziona di nuovo correttamente, MSCS lo rileva e ripassa automaticamente l'applicazione al computer A.

Se si esegue più di un gestore code, considerare la possibilità di avere un disco condiviso separato per ciascuno di essi. Quindi, inserire ciascun gestore code in un gruppo separato in MSCS. In questo modo, qualsiasi gestore code può eseguire il failover sull'altro computer senza influire sugli altri gestori code.

Modalità Attiva / Attiva

In modalità Attiva / Attiva, i computer A e B hanno entrambe applicazioni in esecuzione e i gruppi su ciascun computer sono impostati per utilizzare l'altro computer come backup. Se viene rilevato un errore sul computer A, MSCS trasferisce i dati di stato sul computer B e reinizializza l'applicazione. computer B esegue quindi la propria applicazione e quella di A.

Per questa configurazione sono necessari almeno due dischi condivisi. È possibile configurare MSCS con A come computer preferito per le applicazioni di A e B come computer preferito per le applicazioni di B. Dopo il failover e la riparazione, ogni applicazione finisce automaticamente sul proprio computer.

Per IBM MQ ciò significa che è possibile, ad esempio, eseguire due gestori code, uno su ciascuno di A e B, con ciascuno che sfrutta la potenza completa del proprio computer. Dopo un errore sul computer A, entrambi i gestori code verranno eseguiti sul computer B. Ciò significa condividere la potenza di un computer, con una capacità ridotta di elaborare grandi quantità di dati a velocità. Tuttavia, le applicazioni critiche saranno ancora disponibili mentre si trova e si ripara l'errore su A.

Windows Creazione di un gestore code da utilizzare con MSCS

Questa procedura garantisce che un nuovo gestore code venga creato in modo da essere adatto per la preparazione e il posizionamento sotto il controllo di Microsoft Cluster Service (MSCS).

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Si inizia creando il gestore code con tutte le relative risorse su un'unità locale, quindi si migrano i file di log e i file di dati su un disco condiviso. (È possibile invertire questa operazione.) **Non** tentare di creare un gestore code con le relative risorse su un'unità condivisa.

È possibile creare un gestore code da utilizzare con MSCS in due modi, da un prompt dei comandi o in IBM MQ Explorer. Il vantaggio di utilizzare un prompt dei comandi è che il gestore code viene creato *arrestato* e impostato su *avvio manuale*, che è pronto per MSCS. (Il IBM MQ Explorer avvia automaticamente un nuovo gestore code e lo imposta all'avvio automatico dopo la creazione. Devi cambiare questo.)

Creazione di un gestore code da un prompt dei comandi

Attenersi alla seguente procedura per creare un gestore code da un prompt dei comandi, da utilizzare con MSCS:

1. Assicurarsi che la variabile di ambiente MQSPREFIX sia impostata per fare riferimento a un'unità locale, ad esempio C:\IBM MQ. Se si modifica, riavviare la macchina in modo che l'account di sistema acquisisca la modifica. Se non si imposta la variabile, il gestore code viene creato nella directory predefinita IBM MQ per i gestori code.

2. Creare il gestore code utilizzando il comando **crtmqm** . Ad esempio, per creare un gestore code denominato `mscs_test` nella directory predefinita, utilizzare:

```
crtmqm mscs_test
```

3. Procedere con [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511.

Creazione di un gestore code utilizzando IBM MQ Explorer

Attenersi alla seguente procedura per creare un gestore code utilizzando IBM MQ Explorer, da utilizzare con MSCS:

1. Avviare IBM MQ Explorer dal menu Start.
2. Nella vista Navigator , espandere i nodi della struttura ad albero per trovare i nodi della struttura ad albero `Gestori code` .
3. Fare clic con il pulsante destro del mouse sul nodo `Gestori code` e selezionare **Nuovo > Gestore code**. Viene visualizzato il pannello `Crea gestore code`.
4. Completare la finestra di dialogo (Passo 1), quindi fare clic su **Avanti>**.
5. Completare la finestra di dialogo (Passo 2), quindi fare clic su **Avanti>**.
6. Completare la finestra di dialogo (Passo 3), verificando che `Avvia gestore code` e `Crea canale di connessione server` non siano selezionati, quindi fare clic su **Avanti>**.
7. Completare la finestra di dialogo (passo 4), quindi fare clic su **Fine**.
8. Procedere con [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511.

Windows *Spostamento di un gestore code nella memoria MSCS*

Questa procedura configura un gestore code esistente per renderlo adatto per l'inserimento sotto il controllo di Microsoft Cluster Service (MSCS).

A tale scopo, spostare i file di log e i file di dati su dischi condivisi per renderli disponibili all'altro computer in caso di errore. Ad esempio, il gestore code esistente potrebbe avere percorsi come `C:\WebSphere MQ\Log\QMname` e `C:\WebSphere MQ\qmgrs\QMname`.



Attenzione: Non provare a trasferire i file manualmente; utilizzare il programma di utilità fornito come parte del supporto IBM MQ MSCS come descritto in questo argomento.

Se il gestore code che viene spostato utilizza le connessioni TLS e il repository delle chiavi TLS si trova nella directory dei dati del gestore code sulla macchina locale, il repository delle chiavi verrà spostato con il resto del gestore code sul disco condiviso. Per impostazione predefinita, l'attributo del gestore code che specifica l'ubicazione del repository chiavi TLS, `SSLKEYR`, viene impostato su `MQ_INSTALLATION_PATH\qmgrs\QMGRNAME\ssl\key`, che si trova nella directory dei dati del gestore code. `MQ_INSTALLATION_PATH` rappresenta la directory di livello superiore in cui è installato IBM MQ . Il comando `hamvmqm` non modifica questo attributo del gestore code. In questa situazione è necessario modificare l'attributo del gestore code, `SSLKEYR`, utilizzando il comando IBM MQ Explorer o `MQSC ALTER QMGR`, per puntare al nuovo file del repository delle chiavi TLS.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

La procedura è la seguente:

1. Chiudere il gestore code e controllare che non vi siano errori.
2. Se i file di log del gestore code o i file di coda sono già archiviati su un disco condiviso, saltare il resto di questa procedura e passare direttamente a [“Inserimento di un gestore code sotto il controllo di MSCS”](#) a pagina 512.
3. Eseguire un backup completo dei file di coda e dei file di log e archiviare il backup in un luogo sicuro (consultare [“File di log del gestore code”](#) a pagina 521 per il motivo per cui ciò è importante).

4. Se si dispone già di una risorsa disco condivisa adatta, procedere con il passo 6. In caso contrario, utilizzare MSCS Cluster Administrator per creare una risorsa di tipo *disco condiviso* con capacità sufficiente per memorizzare i file di log del gestore code e i file di dati (coda).
5. Verificare il disco condiviso utilizzando MSCS Cluster Administrator per spostarlo da un nodo cluster all'altro e viceversa.
6. Verificare che il disco condiviso sia in linea sul nodo cluster in cui i file di dati e di log del gestore code sono memorizzati localmente.
7. Eseguire il programma di utilità per spostare il gestore code nel modo seguente:

```
hamvmqm /m qmname /dd " e: \
IBM MQ " /ld " e: \
IBM MQ \log"
```

sostituire il nome del gestore code con *qmname*, la lettera dell'unità disco condivisa per *ee* la directory scelta per *IBM MQ*. Le directory vengono create se non esistono già.

8. Verificare che il gestore code funzioni, utilizzando IBM MQ Explorer. Ad esempio:
 - a. fare clic con il tastino destro del mouse sul nodo della struttura ad albero del gestore code, quindi selezionare **Avvia**. Il gestore code viene avviato.
 - b. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero Code , quindi selezionare **Nuovo > Coda locale ...**, e assegnare un nome alla coda.
 - c. Fare clic su **Fine**.
 - d. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Inserisci messaggio di prova ...**. Viene visualizzato il pannello Inserisci messaggio di prova.
 - e. Immettere del testo del messaggio, quindi fare clic su **Inserisci messaggio di prova** e chiudere il pannello.
 - f. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Sfoggia messaggi ...**. Viene visualizzato il pannello Browser messaggi.
 - g. Assicurarsi che il proprio messaggio sia sulla coda, quindi fare clic su **Chiudi**. Il pannello Browser messaggi viene chiuso.
 - h. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Cancella messaggi ...**. I messaggi sulla coda vengono eliminati.
 - i. Fare clic con il pulsante destro del mouse sulla coda, quindi selezionare **Elimina ...**. Viene visualizzato un pannello di conferma, fare clic su **OK**. La coda è stata eliminata.
 - j. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero del gestore code, quindi selezionare **Arresta ...**. Viene visualizzato il pannello Fine gestore code.
 - k. Fare clic su **OK**. Il gestore code viene arrestato.
9. Come amministratore di IBM MQ , assicurarsi che l'attributo di avvio del gestore code sia impostato su manuale. In IBM MQ Explorer, impostare il campo Avvio su `manua` nel pannello delle proprietà del gestore code.
10. Procedere con [“Inserimento di un gestore code sotto il controllo di MSCS”](#) a pagina 512.

Inserimento di un gestore code sotto il controllo di MSCS

Come inserire un gestore code sotto il controllo di Microsoft Cluster Service (MSCS), incluse le attività prerequisite.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Prima di inserire un gestore code sotto il controllo MSCS/WSFC

Prima di inserire un gestore code sotto il controllo di MSCS/WSFC, completare la seguente procedura:

1. Assicurarsi che IBM MQ e il supporto MSCS/WSFC siano installati su entrambe le macchine nel cluster e che il software su ciascun computer sia identico, come descritto in [“Impostazione di IBM MQ per il clustering MSCS”](#) a pagina 508.
2. Utilizzare il programma di utilità **haregtyp** per registrare IBM MQ come un tipo di risorsa MSCS su tutti i nodi cluster. Consultare [“Supporto per programmi di utilità MSCS”](#) a pagina 523.
3. Se non è stato ancora fatto, [creare un gestore code da utilizzare con MSCS/WSFC](#).
4. Se il gestore code è stato creato o esiste già, assicurarsi di aver eseguito la procedura in [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511.
5. Se il gestore code è in esecuzione, arrestarlo utilizzando un prompt dei comandi o IBM MQ Explorer.
6. Verificare l'operazione MSCS/WSFC delle unità condivise prima di passare a una delle seguenti procedure Windows in questo argomento.

Windows Server 2012, 2016, 2019 o 2022

Per inserire un gestore code sotto il controllo MSCS/WSFC su Windows Server 2012 o versioni successive, utilizzare la seguente procedura:

1. Accedere al computer del nodo cluster che ospita il gestore code oppure accedere a una workstation remota come utente con autorizzazioni di amministrazione del cluster e connettersi al nodo cluster che ospita il gestore code.
2. Avviare lo strumento di gestione cluster di failover.
3. Fare clic con il tasto destro del mouse su **Gestione cluster di failover > Connetti cluster** per aprire una connessione al cluster.
4. A differenza dello schema di gruppi utilizzato in MSCS Cluster Administrator nelle versioni precedenti di Windows, lo strumento Gestione cluster di failover utilizza il concetto di servizi e applicazioni. Un'applicazione o un servizio configurato contiene tutte le risorse necessarie per un'applicazione da raggruppare in cluster. È possibile configurare un gestore code in WSFC nel modo seguente:
 - a. Fare clic con il pulsante destro del mouse sul cluster e selezionare **Configura ruolo** per avviare la configurazione guidata.
 - b. Selezionare **Altro server** nel pannello **Seleziona servizio o applicazione**.
 - c. Selezionare un indirizzo IP appropriato come punto di accesso client.

Questo indirizzo deve essere un indirizzo IP non utilizzato che deve essere utilizzato dai client e da altri gestori code per connettersi al gestore code *virtuale*. Questo indirizzo IP non è l'indirizzo normale (statico) di entrambi i nodi; è un ulteriore indirizzo che *mobile* tra di essi. Sebbene WSFC gestisca l'instradamento di questo indirizzo, **non** verifica che l'indirizzo possa essere raggiunto.

- d. Assegnare una periferica di memoria per l'uso esclusivo da parte del gestore code. Questo dispositivo deve essere creato come istanza di risorsa prima di poter essere assegnato.

È possibile utilizzare un'unità per memorizzare sia i file di log che i file di coda, oppure è possibile suddividerli tra le unità. In entrambi i casi, se ogni gestore code ha il proprio disco condiviso, assicurarsi che tutte le unità utilizzate da questo gestore code siano esclusive per questo gestore code, ossia che nessun altro si basi sulle unità. Assicurarsi inoltre di creare un'istanza di risorsa per ogni unità utilizzata dal gestore code.

Il tipo di risorsa per un'unità dipende dal supporto SCSI che si utilizza; fare riferimento alle istruzioni dell'adattatore SCSI. Potrebbero essere già presenti gruppi e risorse per ciascuna delle unità condivise. In tal caso, non è necessario creare l'istanza della risorsa per ogni unità. Spostarlo dal gruppo corrente a quello creato per il gestore code.

Per ogni risorsa unità, impostare i proprietari possibili su entrambi i nodi. Impostare le risorse dipendenti su nessuna.

- e. Selezionare la risorsa **MQSeries MSCS** sul pannello **Seleziona tipo risorsa**.
 - f. Completare i passi rimanenti nella procedura guidata.
5. Prima di portare la risorsa in linea, la risorsa MQSeries MSCS necessita di ulteriore configurazione:

- a. Selezionare il servizio appena definito che contiene una risorsa denominata 'Nuovo MQSeries MSCS'.
 - b. Fare clic con il tasto destro del mouse su **Proprietà** sulla risorsa IBM MQ .
 - c. Configurare la risorsa:
 - Name ; scegliere un nome che facilita l'identificazione del gestore code a cui è destinato.
 - Run in a separate Resource Monitor ; per un migliore isolamento
 - Possible owners ; imposta entrambi i nodi
 - Dependencies ; aggiungere l'unità e l'indirizzo IP per questo gestore code.

Avviso: Se non si riesce ad aggiungere queste dipendenze, IBM MQ tenta di scrivere lo stato del gestore code sul disco del cluster errato durante i failover. Poiché molti processi potrebbero tentare di scrivere simultaneamente su questo disco, l'esecuzione di alcuni processi IBM MQ potrebbe essere bloccata.

 - Parameters ; come segue:
 - QueueManagerName (obbligatorio); il nome del gestore code che questa risorsa deve controllare. Questo gestore code deve essere presente sul computer locale.
 - PostOnlineCommand (facoltativo); è possibile specificare un programma da eseguire ogni volta che la risorsa del gestore code modifica il proprio stato da non in linea a in linea. Per ulteriori dettagli, fare riferimento a [“Comando PostOnlinee comando PreOfflinein MSCS” a pagina 522.](#)
 - PreOfflineCommand (facoltativo); è possibile specificare un programma da eseguire ogni volta che la risorsa del gestore code modifica il proprio stato da online a offline. Per ulteriori dettagli, fare riferimento a [“Comando PostOnlinee comando PreOfflinein MSCS” a pagina 522.](#)

Nota: L'intervallo di polling *looksAlive* è impostato sul valore predefinito di 5000 ms. L'intervallo di polling *isAlive* è impostato sul valore predefinito di 60 000 ms. Questi valori predefiniti possono essere modificati solo dopo che la definizione della risorsa è stata completata. Per ulteriori dettagli, consultare [“Polling looksAlive e isAlive su MSCS” a pagina 519.](#)
 - d. Facoltativamente, impostare un nodo preferito (ma notare i commenti in [“Utilizzo dei nodi preferiti in MSCS” a pagina 523](#))
 - e. La *Politica di failover* è impostata per default su valori sensibili, ma è possibile ottimizzare le soglie e i periodi che controllano *Failover risorse* e *Failover gruppi* in modo che corrispondano ai carichi collocati sul gestore code.
6. Verificare il gestore code portandolo in linea in MSCS Cluster Administrator e sottoponendolo ad un carico di lavoro di verifica. Se si sta sperimentando con un gestore code di verifica, utilizzare Esplora risorse di IBM MQ . Ad esempio:
 - a. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero Code , quindi selezionare **Nuovo > Coda locale ...**, e assegnare un nome alla coda.
 - b. Fare clic su **Fine**. La coda viene creata e visualizzata nella vista Contenuto.
 - c. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Inserisci messaggio di prova ...**. Viene visualizzato il pannello Inserisci messaggio di prova.
 - d. Immettere del testo del messaggio, quindi fare clic su **Inserisci messaggio di prova** e chiudere il pannello.
 - e. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Sfoglia messaggi ...**. Viene visualizzato il pannello Browser messaggi.
 - f. Assicurarsi che il proprio messaggio sia nella coda, quindi fare clic su **Chiudi**. Il pannello Browser messaggi viene chiuso.
 - g. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Cancella messaggi ...**. I messaggi sulla coda vengono eliminati.

- h. Fare clic con il pulsante destro del mouse sulla coda, quindi selezionare **Elimina** Viene visualizzato un pannello di conferma, fare clic su **OK**. La coda è stata eliminata.
7. Verificare che il gestore code possa essere portato offline e di nuovo online utilizzando MSCS Cluster Administrator.
8. Simula un failover.

In MSCS Cluster Administrator, fare clic con il tasto destro del mouse sul gruppo contenente il gestore code e selezionare **Move Group**. Questa operazione può richiedere alcuni minuti. Se in altre occasioni si desidera spostare rapidamente un gestore code su un altro nodo, seguire la procedura in ["Spostamento di un gestore code nella memoria MSCS"](#) a pagina 511. È anche possibile fare clic con il pulsante destro del mouse e selezionare **Initiate Failure**; l'azione (riavvio locale o failover) dipende dallo stato corrente e dalle impostazioni di configurazione.

Server Windows 2008

Per posizionare un gestore code sotto il controllo di MSCS su Windows Server 2008, utilizzare la procedura riportata di seguito:

1. Accedere al computer del nodo cluster che ospita il gestore code oppure accedere a una workstation remota come utente con autorizzazioni di amministrazione del cluster e connettersi al nodo cluster che ospita il gestore code.
2. Avviare lo strumento di gestione cluster di failover.
3. Fare clic con il tasto destro del mouse su **Gestione cluster di failover > Gestisci cluster ...** per aprire una connessione al cluster.
4. A differenza dello schema di gruppi utilizzato in MSCS Cluster Administrator nelle versioni precedenti di Windows, lo strumento Gestione cluster di failover utilizza il concetto di servizi e applicazioni. Un'applicazione o un servizio configurato contiene tutte le risorse necessarie per un'applicazione da raggruppare in cluster. È possibile configurare un gestore code in MSCS nel modo seguente:
 - a. Fare clic con il tasto destro del mouse su **Servizi e applicazioni > Configura un servizio o un'applicazione ...** per avviare la procedura guidata di configurazione.
 - b. Selezionare **Altro server** nel pannello **Seleziona servizio o applicazione**.
 - c. Selezionare un indirizzo IP appropriato come punto di accesso client.

Questo indirizzo deve essere un indirizzo IP non utilizzato che deve essere utilizzato dai client e da altri gestori code per connettersi al gestore code *virtuale*. Questo indirizzo IP non è l'indirizzo normale (statico) di entrambi i nodi; è un ulteriore indirizzo che *mobile* tra di essi. Sebbene MSCS gestisca l'instradamento di questo indirizzo, **non** verifica che l'indirizzo possa essere raggiunto.

- d. Assegnare una periferica di memoria per l'uso esclusivo da parte del gestore code. Questo dispositivo deve essere creato come istanza di risorsa prima di poter essere assegnato.

È possibile utilizzare un'unità per memorizzare sia i file di log che i file di coda, oppure è possibile suddividerli tra le unità. In entrambi i casi, se ogni gestore code ha il proprio disco condiviso, assicurarsi che tutte le unità utilizzate da questo gestore code siano esclusive per questo gestore code, ossia che nessun altro si basi sulle unità. Assicurarsi inoltre di creare un'istanza di risorsa per ogni unità utilizzata dal gestore code.

Il tipo di risorsa per un'unità dipende dal supporto SCSI che si utilizza; fare riferimento alle istruzioni dell'adattatore SCSI. Potrebbero essere già presenti gruppi e risorse per ciascuna delle unità condivise. In tal caso, non è necessario creare l'istanza della risorsa per ogni unità. Spostarlo dal gruppo corrente a quello creato per il gestore code.

Per ogni risorsa unità, impostare i proprietari possibili su entrambi i nodi. Impostare le risorse dipendenti su nessuna.

- e. Selezionare la risorsa **MQSeries MSCS** sul pannello **Seleziona tipo risorsa**.
 - f. Completare i passi rimanenti nella procedura guidata.
5. Prima di portare la risorsa in linea, la risorsa MQSeries MSCS necessita di ulteriore configurazione:

- a. Selezionare il nuovo servizio definito che contiene una risorsa denominata 'Nuovo MQSeries MSCS'.
 - b. Fare clic con il tasto destro del mouse su **Proprietà** nella risorsa MQ .
 - c. Configurare la risorsa:
 - Name ; scegliere un nome che facilita l'identificazione del gestore code a cui è destinato.
 - Run in a separate Resource Monitor ; per un migliore isolamento
 - Possible owners ; imposta entrambi i nodi
 - Dependencies ; aggiungere l'unità e l'indirizzo IP per questo gestore code.

Avviso: Se non si riesce ad aggiungere queste dipendenze, IBM MQ tenta di scrivere lo stato del gestore code sul disco del cluster errato durante i failover. Poiché molti processi potrebbero tentare di scrivere simultaneamente su questo disco, l'esecuzione di alcuni processi IBM MQ potrebbe essere bloccata.

 - Parameters ; come segue:
 - QueueManagerName (obbligatorio); il nome del gestore code che questa risorsa deve controllare. Questo gestore code deve essere presente sul computer locale.
 - PostOnlineCommand (facoltativo); è possibile specificare un programma da eseguire ogni volta che la risorsa del gestore code modifica il proprio stato da non in linea a in linea. Per ulteriori dettagli, fare riferimento a [“Comando PostOnlinee comando PreOfflinein MSCS” a pagina 522.](#)
 - PreOfflineCommand (facoltativo); è possibile specificare un programma da eseguire ogni volta che la risorsa del gestore code modifica il proprio stato da online a offline. Per ulteriori dettagli, fare riferimento a [“Comando PostOnlinee comando PreOfflinein MSCS” a pagina 522.](#)

Nota: L'intervallo di polling *looksAlive* è impostato sul valore predefinito di 5000 ms. L'intervallo di polling *isAlive* è impostato sul valore predefinito di 60 000 ms. Questi valori predefiniti possono essere modificati solo dopo che la definizione della risorsa è stata completata. Per ulteriori dettagli, consultare [“Polling looksAlive e isAlive su MSCS” a pagina 519.](#)
 - d. Facoltativamente, impostare un nodo preferito (ma notare i commenti in [“Utilizzo dei nodi preferiti in MSCS” a pagina 523](#))
 - e. La *Politica di failover* è impostata per default su valori sensibili, ma è possibile ottimizzare le soglie e i periodi che controllano *Failover risorse* e *Failover gruppi* in modo che corrispondano ai carichi collocati sul gestore code.
6. Verificare il gestore code portandolo in linea in MSCS Cluster Administrator e sottoponendolo ad un carico di lavoro di verifica. Se si sta sperimentando con un gestore code di verifica, utilizzare Esplora risorse di IBM MQ . Ad esempio:
 - a. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero Code , quindi selezionare **Nuovo > Coda locale ...**, e assegnare un nome alla coda.
 - b. Fare clic su **Fine**. La coda viene creata e visualizzata nella vista Contenuto.
 - c. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Inserisci messaggio di prova ...**. Viene visualizzato il pannello **Inserisci messaggio di prova** .
 - d. Immettere del testo del messaggio, quindi fare clic su **Inserisci messaggio di prova** e chiudere il pannello.
 - e. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Sfoggia messaggi ...**. Viene visualizzato il pannello **Browser messaggi** .
 - f. Assicurarsi che il proprio messaggio sia nella coda, quindi fare clic su **Chiudi**. Il pannello **Browser messaggi** viene chiuso.
 - g. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Cancella messaggi ...**. I messaggi sulla coda vengono eliminati.
 - h. Fare clic con il pulsante destro del mouse sulla coda, quindi selezionare **Elimina ...**. Viene visualizzato un pannello di conferma, fare clic su **OK**. La coda è stata eliminata.

7. Verificare che il gestore code possa essere portato offline e di nuovo online utilizzando MSCS Cluster Administrator.
8. Simula un failover.

In MSCS Cluster Administrator, fare clic con il tasto destro del mouse sul gruppo contenente il gestore code e selezionare *Move Group*. Questa operazione può richiedere alcuni minuti. Se in altre occasioni si desidera spostare rapidamente un gestore code su un altro nodo, seguire la procedura in [“Spostamento di un gestore code nella memoria MSCS” a pagina 511](#). È anche possibile fare clic con il pulsante destro del mouse e selezionare *Initiate Failure*; l'azione (riavvio locale o failover) dipende dallo stato corrente e dalle impostazioni di configurazione.

Windows 2003

Per posizionare un gestore code sotto il controllo di MSCS su Windows 2003, utilizzare la seguente procedura:

1. Accedere al computer del nodo cluster che ospita il gestore code oppure accedere a una workstation remota come utente con autorizzazioni di amministrazione del cluster e connettersi al nodo cluster che ospita il gestore code.
2. Avviare MSCS Cluster Administrator.
3. Aprire una connessione al cluster.
4. Creare un gruppo MSCS da utilizzare per contenere le risorse per il gestore code. Denominare il gruppo in modo tale che sia ovvio a quale gestore code è correlato. Ogni gruppo può contenere più gestori code, come descritto in [“Utilizzo di più gestori code con MSCS” a pagina 509](#).

Utilizzare il gruppo per tutti i passi rimanenti.

5. Creare un'istanza di risorsa per ciascuna unità logica SCSI utilizzata dal gestore code.

È possibile utilizzare un'unità per memorizzare sia i file di log che i file di coda, oppure è possibile suddividerli tra le unità. In entrambi i casi, se ogni gestore code ha il proprio disco condiviso, assicurarsi che tutte le unità utilizzate da questo gestore code siano esclusive per questo gestore code, ossia che nessun altro si basi sulle unità. Assicurarsi inoltre di creare un'istanza di risorsa per ogni unità utilizzata dal gestore code.

Il tipo di risorsa per un'unità dipende dal supporto SCSI che si utilizza; fare riferimento alle istruzioni dell'adattatore SCSI. Potrebbero essere già presenti gruppi e risorse per ciascuna delle unità condivise. In tal caso, non è necessario creare l'istanza della risorsa per ogni unità. Spostarlo dal gruppo corrente a quello creato per il gestore code.

Per ogni risorsa unità, impostare i proprietari possibili su entrambi i nodi. Impostare le risorse dipendenti su nessuna.

6. Creare un'istanza di risorsa per l'indirizzo IP.

Creare una risorsa indirizzo IP (tipo di risorsa *indirizzo IP*). Questo indirizzo deve essere un indirizzo IP non utilizzato che deve essere utilizzato dai client e da altri gestori code per connettersi al gestore code *virtuale*. Questo indirizzo IP non è l'indirizzo normale (statico) di entrambi i nodi; è un ulteriore indirizzo che *mobile* tra di essi. Sebbene MSCS gestisca l'instradamento di questo indirizzo, **non** verifica che l'indirizzo possa essere raggiunto.

7. Creare un'istanza di risorsa per il gestore code.

Creare una risorsa di tipo *IBM MQ MSCS*. La procedura guidata richiede diversi elementi, tra cui:

- *Name*; scegliere un nome che facilita l'identificazione del gestore code a cui è destinato.
- *Add to group*; utilizzare il gruppo creato
- *Run in a separate Resource Monitor*; per un migliore isolamento
- *Possible owners*; imposta entrambi i nodi
- *Dependencies*; aggiungere l'unità e l'indirizzo IP per questo gestore code.

Avviso: Se non si riesce ad aggiungere queste dipendenze, IBM MQ tenta di scrivere lo stato del gestore code sul disco del cluster errato durante i failover. Poiché molti processi potrebbero tentare di scrivere simultaneamente su questo disco, l'esecuzione di alcuni processi IBM MQ potrebbe essere bloccata.

• Parameters ; come segue:

- QueueManagerName (obbligatorio); il nome del gestore code che questa risorsa deve controllare. Questo gestore code deve essere presente sul computer locale.
- PostOnlineCommand (facoltativo); è possibile specificare un programma da eseguire ogni volta che la risorsa del gestore code modifica il proprio stato da non in linea a in linea. Per ulteriori dettagli, fare riferimento a [“Comando PostOnlinee comando PreOfflinein MSCS”](#) a pagina 522.
- PreOfflineCommand (facoltativo); è possibile specificare un programma da eseguire ogni volta che la risorsa del gestore code modifica il proprio stato da online a offline. Per ulteriori dettagli, fare riferimento a [“Comando PostOnlinee comando PreOfflinein MSCS”](#) a pagina 522.

Nota: L'intervallo di polling *looksAlive* è impostato sul valore predefinito di 5000 ms. L'intervallo di polling *isAlive* è impostato sul valore predefinito di 30000 ms. Questi valori predefiniti possono essere modificati solo dopo che la definizione della risorsa è stata completata. Per ulteriori dettagli, consultare [“Polling looksAlive e isAlive su MSCS”](#) a pagina 519.

8. Facoltativamente, impostare un nodo preferito (ma notare i commenti in [“Utilizzo dei nodi preferiti in MSCS”](#) a pagina 523)
9. La *Politica di failover* (come definita nelle proprietà per il gruppo) è impostata per impostazione predefinita su valori sensibili, ma è possibile ottimizzare le soglie e i periodi che controllano *Failover risorse* e *Failover gruppi* in modo che corrispondano ai caricamenti posizionati sul gestore code.
10. Verificare il gestore code portandolo in linea in MSCS Cluster Administrator e sottoponendolo ad un carico di lavoro di verifica. Se si sta sperimentando con un gestore code di verifica, utilizzare Esplora risorse di IBM MQ . Ad esempio:
 - a. Fare clic con il pulsante destro del mouse sul nodo della struttura ad albero Code , quindi selezionare **Nuovo > Coda locale ...**, e assegnare un nome alla coda.
 - b. Fare clic su **Fine**. La coda viene creata e visualizzata nella vista Contenuto.
 - c. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Inserisci messaggio di prova** Viene visualizzato il pannello **Inserisci messaggio di prova** .
 - d. Immettere del testo del messaggio, quindi fare clic su **Inserisci messaggio di prova** e chiudere il pannello.
 - e. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Sfoglia messaggi** Viene visualizzato il pannello **Browser messaggi** .
 - f. Assicurarsi che il proprio messaggio sia nella coda, quindi fare clic su **Chiudi**. Il pannello **Browser messaggi** viene chiuso.
 - g. Fare clic con il tasto destro del mouse sulla coda, quindi selezionare **Cancella messaggi** I messaggi sulla coda vengono eliminati.
 - h. Fare clic con il pulsante destro del mouse sulla coda, quindi selezionare **Elimina** Viene visualizzato un pannello di conferma, fare clic su **OK**. La coda è stata eliminata.
11. Verificare che il gestore code possa essere portato offline e di nuovo online utilizzando MSCS Cluster Administrator.
12. Simula un failover.

In MSCS Cluster Administrator, fare clic con il tasto destro del mouse sul gruppo contenente il gestore code e selezionare *Move Group*. Questa operazione può richiedere alcuni minuti. Se in altre occasioni si desidera spostare rapidamente un gestore code su un altro nodo, seguire la procedura in [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511. È anche possibile fare clic con il pulsante destro del mouse e selezionare *Initiate Failure* ; l'azione (riavvio locale o failover) dipende dallo stato corrente e dalle impostazioni di configurazione.

Windows **Polling looksAlive e isAlive su MSCS**

looksAlive e *isAlive* sono intervalli in cui Microsoft Cluster Service (MSCS) richiama il codice della libreria fornito dei tipi di risorsa e richiede che la risorsa esegua controlli per determinare lo stato di funzionamento di se stessa. Ciò determina se MSCS tenta di eseguire il failover della risorsa.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Ogni volta che trascorre l'intervallo *looksAlive* (valore predefinito 5000 ms), la risorsa del gestore code viene richiamata per eseguire il proprio controllo per determinare se il suo stato è soddisfacente.

Ogni volta che trascorre l'intervallo *isAlive* (valore predefinito 30000 ms), viene effettuata un'altra chiamata alla risorsa del gestore code per eseguire un altro controllo per determinare se la risorsa sta funzionando correttamente. Ciò abilita due livelli di controllo del tipo di risorsa.

1. Un controllo di stato *looksAlive* per stabilire se la risorsa sembra funzionare.
2. Un controllo *isAlive* più significativo che determina se la risorsa del gestore code è attiva.

Se si determina che la risorsa del gestore code non è attiva, MSCS, in base ad altre opzioni MSCS avanzate, attiva un failover per la risorsa e le risorse dipendenti associate ad un altro nodo nel cluster. Per ulteriori informazioni, consultare la [documentazione MSCS](#).

Windows **Rimozione di un gestore code dal controllo MSCS**

È possibile rimuovere i gestori code dal controllo Microsoft Cluster Service (MSCS) e riportarli alla gestione manuale.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Non è necessario rimuovere i gestori code dal controllo MSCS per le operazioni di manutenzione. È possibile farlo portando un gestore code offline temporaneamente, utilizzando MSCS Cluster Administrator. La rimozione di un gestore code dal controllo MSCS è una modifica più permanente; eseguire tale operazione solo se si decide che non si desidera più che MSCS disponga di un ulteriore controllo del gestore code.

Se il gestore code viene rimosso utilizza connessioni TSL, è necessario modificare l'attributo del gestore code, SSLKEYR, utilizzando il comando IBM MQ Explorer o MQSC ALTER QMGR, per puntare al file del repository delle chiavi TLS nella directory locale.

La procedura è:

1. Portare offline la risorsa del gestore code utilizzando MSCS Cluster Administrator, come descritto in [“Portare un gestore code non in linea da MSCS”](#) a pagina 519
2. Eliminare l'istanza di risorsa. Questa operazione non elimina il gestore code.
3. Facoltativamente, migrare di nuovo i file del gestore code dalle unità condivise alle unità locali. Per fare ciò, consultare [“Restituzione di un gestore code dalla memoria MSCS”](#) a pagina 520.
4. Verificare il gestore code.

Portare un gestore code non in linea da MSCS

Per disattivare un gestore code da MSCS, effettuare le seguenti operazioni:

1. Avviare MSCS Cluster Administrator.
2. Aprire una connessione al cluster.
3. Selezionare Groupso Role se si sta utilizzando Windows 2012 e aprire il gruppo contenente il gestore code da spostare.
4. Selezionare la risorsa gestore code.
5. Fare clic con il pulsante destro del mouse e selezionare Offline.
6. Attendere il completamento.

Restituzione di un gestore code dalla memoria MSCS

Questa procedura configura il gestore code in modo che sia di nuovo sull'unità locale del computer, ovvero diventa un gestore code *normale* IBM MQ. A tale scopo, spostare i file di log e i file di dati dai dischi condivisi. Ad esempio, il gestore code esistente potrebbe avere percorsi come E:\WebSphere MQ\log\QMname e E:\WebSphere MQ\qmgrs\QMname. Non provare a spostare i file manualmente; utilizzare il programma di utilità **hamvmqm** fornito come parte del supporto IBM MQ MSCS:

1. Eseguire un backup completo dei file di coda e dei file di log e archiviare il backup in un luogo sicuro (consultare [“File di log del gestore code”](#) a pagina 521 per il motivo per cui ciò è importante).
2. Decidere quale unità locale utilizzare e verificare che disponga di capacità sufficiente per memorizzare i file di log del gestore code e i file di dati (coda).
3. Assicurarsi che il disco condiviso su cui si trovano attualmente i file sia in linea sul nodo cluster in cui spostare i file di dati e di log del gestore code.
4. Eseguire il programma di utilità per spostare il gestore code nel modo seguente:

```
hamvmqm /m qmname /dd " c:\
IBM MQ " /ld "c:\
IBM MQ \log"
```

sostituire il nome del gestore code con *qmname*, la lettera dell'unità disco locale per ce la directory scelta per *IBM MQ* (le directory vengono create se non esistono già).

5. Verificare il funzionamento del gestore code (come descritto in [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511).

Windows *Suggerimenti e consigli sull'utilizzo di MSCS*

Questa sezione contiene alcune informazioni generali che consentono di utilizzare in modo efficace il supporto IBM MQ per Microsoft Cluster Service (MSCS).

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Quanto tempo ci vuole per far fallire un gestore code da una macchina all'altra? Ciò dipende fortemente dalla quantità di carico di lavoro sul gestore code e dalla combinazione di traffico, ad esempio, la quantità di traffico persistente, all'interno del punto di sincronizzazione e la quantità di commit prima dell'errore. I test IBM hanno fornito tempi di failover e failback di circa un minuto. Questo si trovava su un gestore code caricato in modo molto leggero e i tempi effettivi variano notevolmente a seconda del carico.

Windows *Verifica del funzionamento di MSCS*

Seguire questa procedura per assicurarsi di avere un cluster MSCS in esecuzione.

Le descrizioni delle attività che iniziano con [“Creazione di un gestore code da utilizzare con MSCS”](#) a pagina 510 presuppongono che si disponga di un cluster MSCS in esecuzione in cui è possibile creare, migrare ed eliminare le risorse. Se si desidera assicurarsi di disporre di un cluster di questo tipo:

1. Utilizzando MSCS Cluster Administrator, creare un gruppo.
2. All'interno di tale gruppo, creare una istanza di una risorsa dell'applicazione generica, specificando l'orologio di sistema (nome percorso C:\winnt\system32\clock.exe e directory di lavoro C:\).
3. Assicurarsi di poter portare la risorsa in linea, di poter spostare il gruppo che la contiene sull'altro nodo e di poter portare la risorsa fuori linea.

Windows *Avvio manuale e MSCS*

Per un gestore code gestito da MSCS, è necessario impostare l'attributo di avvio su manuale. Ciò garantisce che il supporto MSCS IBM MQ possa riavviare MQSeries Service senza avviare immediatamente il gestore code.

Il supporto MSCS di IBM MQ deve essere in grado di riavviare il servizio in modo che possa eseguire il monitoraggio e il controllo, ma deve rimanere esso stesso in controllo di quali gestori code sono in

esecuzione e su quali macchine. Per ulteriori informazioni, consultare [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511.

MSCS ed i gestori code

Considerazioni relative ai gestori code quando si utilizza MSCS.

Creazione di un gestore code corrispondente sull'altro nodo

Perché il clustering funzioni con IBM MQ, è necessario un gestore code identico sul nodo B per ogni gestore code sul nodo A. Tuttavia, non è necessario creare esplicitamente la seconda. È possibile creare o preparare un gestore code su un nodo, spostarlo sull'altro nodo come descritto in [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511 ed è completamente duplicato su tale nodo.

Gestori code predefiniti

Non utilizzare un gestore code predefinito sotto il controllo di MSCS. Un gestore code non dispone di una proprietà che lo rende predefinito; IBM MQ conserva il proprio record separato. Se si sposta un set di gestori code in modo che sia il valore predefinito sull'altro computer in failover, non diventa il valore predefinito. Tutte le applicazioni fanno riferimento a gestori code specifici in base al nome.

Eliminazione di un gestore code

Una volta che un gestore code ha spostato il nodo, i relativi dettagli sono presenti nel registro su entrambi i computer. Quando si desidera eliminarlo, eseguire tale operazione normalmente su un computer, quindi eseguire il programma di utilità descritto in [“Supporto per programmi di utilità MSCS”](#) a pagina 523 per ripulire il registro sull'altro computer.

Supporto per i gestori code esistenti

È possibile inserire un gestore code esistente sotto il controllo di MSCS, purché sia possibile inserire i file di log del gestore code e i file di coda su un disco che si trova sul bus SCSI condiviso tra le due macchine (consultare [Figura 71](#) a pagina 508). È necessario disattivare brevemente il gestore code durante la creazione della risorsa MSCS.

Se si desidera creare un nuovo gestore code, crearlo indipendentemente da MSCS, verificarlo, quindi metterlo sotto controllo MSCS. Consultare:

- [“Creazione di un gestore code da utilizzare con MSCS”](#) a pagina 510
- [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511
- [“Inserimento di un gestore code sotto il controllo di MSCS”](#) a pagina 512

Come comunicare a MSCS quali gestori code gestire

È possibile scegliere quali gestori code vengono posti sotto il controllo di MSCS utilizzando MSCS Cluster Administrator per creare un'istanza della risorsa per ciascun gestore code. Questo processo presenta un elenco di risorse da cui selezionare il gestore code che si desidera venga gestito da tale istanza.

File di log del gestore code

Quando si sposta un gestore code nella memoria MSCS, si spostano i relativi file di log e di dati su un disco condiviso (ad esempio, consultare [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511).

Si consiglia di chiudere il gestore code in modo pulito e di eseguire un backup completo dei file di dati e dei file di log.

Più gestori code

IBM MQ Il supporto MSCS consente di eseguire più gestori code su ciascuna macchina e di porre singoli gestori code sotto il controllo di MSCS.

Windows *Utilizza sempre MSCS per gestire i cluster*

Non tentare di eseguire operazioni di avvio e di arresto direttamente su qualsiasi gestore code sotto il controllo di MSCS, utilizzando i comandi di controllo o IBM MQ Explorer. Utilizzare invece MSCS Cluster Administrator per portare il gestore code online o offline.

L'utilizzo di MSCS Cluster Administrator è in parte per evitare possibili confusioni causate dal fatto che MSCS riporta che il gestore code è offline, quando in realtà è stato avviato al di fuori del controllo di MSCS. Più seriamente, l'arresto di un gestore code senza utilizzare MSC viene rilevato da MSCS come un errore, avviando il failover sull'altro nodo.

Windows *Lavorare in modalità Attiva / Attiva in MSCS*

Entrambi i computer nel cluster MSCS possono eseguire gestori code in modalità Attiva / Attiva. Non è necessario avere una macchina completamente inattiva che funge da standby (ma è possibile, se si desidera, in modalità Attiva / Passiva).

Se si intende utilizzare entrambe le macchine per eseguire il carico di lavoro, fornire a ciascuna di esse una capacità sufficiente (processore, memoria, memoria secondaria) per eseguire l'intero carico di lavoro del cluster ad un livello di prestazioni soddisfacente.

Nota: Se si utilizza MSCS insieme a Microsoft Transaction Server (COM +), **non è possibile** utilizzare la modalità Attiva / Attiva. Questo perché, per utilizzare IBM MQ con MSCS e COM +:

- I componenti dell'applicazione che utilizzano il supporto IBM MQ COM + devono essere eseguiti sullo stesso computer del DTC (Distributed Transaction Coordinator), una parte di COM +.
- Il gestore code deve essere eseguito anche sullo stesso computer.
- Il DTC deve essere configurato come una risorsa MSCS e può quindi essere eseguito solo su uno dei computer nel cluster in qualsiasi momento.

Windows *Comando PostOnlinee comando PreOfflinein MSCS*

Utilizzare questi comandi per integrare il supporto IBM MQ MSCS con altri sistemi. È possibile utilizzarli per immettere comandi IBM MQ , con alcune limitazioni.

Specificare questi comandi nei parametri per una risorsa di tipo IBM MQ MSCS. È possibile utilizzarli per integrare il supporto MSCS IBM MQ con altri sistemi o procedure. Ad esempio, è possibile specificare il nome di un programma che invia un messaggio di posta, attiva un cercapersone o genera un'altra forma di avviso da catturare da un altro sistema di controllo.

Il comando PostOnlineviene richiamato quando la risorsa passa da offline a online; il comando PreOfflineviene richiamato per una modifica da online a offline. Quando richiamati questi comandi vengono eseguiti, per impostazione predefinita, dalla directory di sistema Windows . Poiché IBM MQ utilizza un processo di controllo delle risorse a 32 bit, su sistemi Windows a 64 bit, questa è la directory \Windows\SysWOW64 anziché la directory \Windows\system32 . Per ulteriori informazioni, consultare la documentazione Microsoft sul reindirizzamento dei file in un ambiente Windows x64 . Entrambi i comandi vengono eseguiti con l'account utente utilizzato per eseguire il Servizio cluster MSCS; e vengono richiamati in modo asincrono; il supporto IBM MQ MSCS non li attende per il completamento prima di continuare. Ciò elimina qualsiasi rischio che possano bloccare o ritardare ulteriori operazioni del cluster.

È anche possibile utilizzare questi comandi per immettere comandi IBM MQ , ad esempio per riavviare i canali del richiedente. Tuttavia, i comandi vengono eseguiti nel momento in cui lo stato del gestore code cambia in modo che non siano destinati ad eseguire funzioni di lunga durata e non devono fare ipotesi sullo stato corrente del gestore code; è possibile che, immediatamente dopo che il gestore code è stato portato in linea, un amministratore abbia emesso un comando non in linea.

Se si desidera eseguire programmi che dipendono dallo stato del gestore code, creare istanze del tipo di risorsa MSCS *Generic Application*, collocarle nello stesso gruppo MSCS della risorsa del gestore code e renderle dipendenti dalla risorsa del gestore code.

Windows *Utilizzo dei nodi preferiti in MSCS*

Può essere utile quando si utilizza la modalità Attiva / Attiva in MSCS per configurare un *nodo preferito* per ciascun gestore code. Tuttavia, in generale, è meglio non impostare un nodo preferito, ma affidarsi a un failback manuale.

A differenza di alcune altre risorse relativamente stateless, un gestore code può impiegare del tempo per eseguire il failover (o il backover) da un nodo all'altro. Per evitare interruzioni non necessarie, verificare il nodo ripristinato prima di ripristinare un gestore code. Ciò impedisce l'utilizzo dell'impostazione di failback *immediate*. È possibile configurare il failback in modo che si verifichi tra determinate ore del giorno.

Probabilmente l'instradamento più sicuro consiste nello spostare il gestore code manualmente sul nodo richiesto, quando si è certi che il nodo è completamente ripristinato. Ciò impedisce l'utilizzo dell'opzione *preferred node*.

Windows *Errori COM + durante l'installazione su MSCS*

Quando si installa IBM MQ su un cluster MSCS appena installato, è possibile che venga rilevato un errore con COM + di origine e ID evento 4691 riportato nel log eventi dell'applicazione.

Ciò significa che si sta tentando di eseguire IBM MQ su un ambiente MSCS (Microsoft Cluster Server) quando Microsoft Distributed Transaction Coordinator (MSDTC) non è stato configurato per l'esecuzione in tale ambiente. Per informazioni sulla configurazione di MSDTC in un ambiente con cluster, fare riferimento alla documentazione Microsoft.

Windows *Supporto per programmi di utilità MSCS*

Un elenco del supporto IBM MQ per i programmi di programma di utilità Microsoft Cluster Service (MSCS) che è possibile eseguire da una richiesta comandi.

Nota: Da Windows Server 2016, il nuovo nome per Microsoft Cluster Service (MSCS) è Windows Server Failover Clustering (WSFC).

Il supporto IBM MQ per MSCS include i seguenti programmi di utilità:

Registrazione/Annullamento della registrazione del tipo di risorsa

`haregtyp.exe`

Dopo aver *annullato la registrazione* del tipo di risorsa MSCS IBM MQ, non è più possibile creare risorse di quel tipo. MSCS non consente di annullare la registrazione di un tipo di risorsa se si dispone ancora di istanze di quel tipo nel cluster:

1. Utilizzando l'amministratore del cluster MSCS, arrestare tutti i gestori code in esecuzione sotto il controllo di MSCS, portandoli offline come descritto in ["Portare un gestore code non in linea da MSCS"](#) a pagina 519.
2. Utilizzando MSCS Cluster Administrator, eliminare le istanze della risorsa.
3. Al prompt dei comandi, annullare la registrazione del tipo di risorsa immettendo il seguente comando:

```
haregtyp /u
```

Se si desidera *registrare* il tipo (o registrarlo di nuovo in un secondo momento), immettere il seguente comando da un prompt dei comandi:

```
haregtyp /r
```

Dopo aver registrato correttamente le librerie MSCS, è necessario riavviare il sistema se non lo si è fatto dall'installazione di IBM MQ.

Spostare un gestore code nella memoria MSCS

hamvmqm.exe

Consultare [“Spostamento di un gestore code nella memoria MSCS”](#) a pagina 511.

Eliminare un gestore code da un nodo

hadlrmqm.exe

Considera il caso in cui hai avuto un gestore code nel tuo cluster, è stato spostato da un nodo a un altro e ora vuoi distruggerlo. Utilizzare Esplora risorse di IBM MQ per eliminarlo sul nodo in cui si trova attualmente. Le voci di registro per esso esistono ancora sull'altro computer. Per eliminarli, immettere il seguente comando in un prompt su tale computer:

```
hadlrmqm /m qmname
```

dove qmname è il nome del gestore code da eliminare.

Controllare e salvare i dettagli di configurazione

amqmsysn.exe

Questo programma di utilità presenta una finestra di dialogo che mostra i dettagli completi dell'impostazione del supporto MSCS IBM MQ, come potrebbe essere richiesto se si chiama il supporto IBM. È disponibile un'opzione per salvare i dettagli in un file.

Multi

Gestori code a più istanze

I gestori code a più istanze sono istanze dello stesso gestore code configurato su server differenti. Un'istanza del gestore code è definita come istanza attiva e un'istanza è definita come istanza in standby. Se l'istanza attiva ha esito negativo, il gestore code a più istanze viene riavviato automaticamente sul server di standby.

Esempio di configurazione del gestore code a più istanze

Figura 72 a pagina 525 mostra un esempio di configurazione a più istanze per il gestore code QM1. IBM MQ è installato su due server, uno dei quali è di riserva. È stato creato un gestore code, QM1. Un'istanza di QM1 è attiva ed è in esecuzione su un server. L'altra istanza di QM1 è in esecuzione in standby sull'altro server, non eseguendo alcuna elaborazione attiva, ma è pronta a subentrare all'istanza attiva di QM1, se l'istanza attiva ha esito negativo. (Ci può essere solo un'istanza attiva e un'istanza in standby del gestore code in una configurazione a più istanze.)

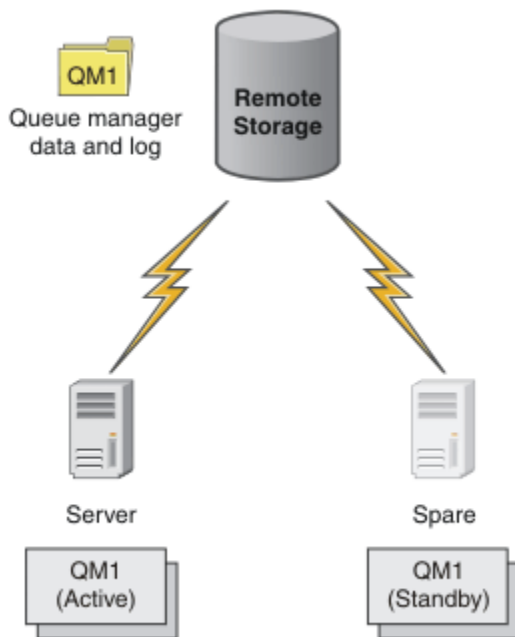


Figura 72. gestore code a più istanze

Quando si intende utilizzare un gestore code come gestore code a più istanza, creare un singolo gestore code su uno dei server utilizzando il comando **crtmqm**, inserendo i relativi dati del gestore code e log nella memoria di rete condivisa. Sull'altro server, piuttosto che creare nuovamente il gestore code, utilizzare il comando **addmqinf** per creare un riferimento ai log e ai dati del gestore code nella memoria di rete.

Ora è possibile eseguire il gestore code da entrambi i server. Ciascuno dei server fa riferimento agli stessi dati e log del gestore code; esiste un solo gestore code ed è attivo su un solo server alla volta.

Il gestore code può essere eseguito come gestore code a istanza singola o come gestore code a più istanze. In entrambi i casi è in esecuzione una sola istanza del gestore code, che elabora le richieste. La differenza è che quando viene eseguito come gestore code a più istanze, il server che non sta eseguendo l'istanza attiva del gestore code viene eseguito come un'istanza in standby, pronta a prendere il posto dell'istanza attiva automaticamente se il server attivo ha esito negativo.

L'unico controllo che si ha su quale istanza diventa attiva per prima è l'ordine in cui si avvia il gestore code sui due server. La prima istanza per acquisire i blocchi di lettura / scrittura sui dati del gestore code diventa l'istanza attiva.

È possibile scambiare l'istanza attiva con l'altro server, una volta avviata, arrestando l'istanza attiva utilizzando l'opzione di commutazione per trasferire il controllo allo standby.

L'istanza attiva di QM1 ha accesso esclusivo ai dati del gestore code condiviso e alle cartelle di log quando è in esecuzione. L'istanza standby di QM1 rileva quando l'istanza attiva ha avuto esito negativo e diventa l'istanza attiva. Assume il controllo dei dati e dei log QM1 nello stato in cui sono stati lasciati dall'istanza attiva e accetta le riconessioni da client e canali.

L'istanza attiva potrebbe avere esito negativo per vari motivi che determinano l'assunzione del controllo da parte dello standby:

- Errore del server che ospita l'istanza del gestore code attivo.
- Errore di connettività tra il server che ospita l'istanza del gestore code attivo e il filesystem.
- Mancata risposta dei processi del gestore code, rilevata da IBM MQ, che quindi arresta il gestore code.

È possibile aggiungere le informazioni di configurazione del gestore code a più server e scegliere due server qualsiasi da eseguire come coppia attivo / standby. Esiste un limite di un totale di due istanze. Non è possibile avere due istanze in standby e una attiva.

Componenti aggiuntivi necessari per creare una soluzione alta disponibilità

Un gestore code a più istanze è una parte di una soluzione alta disponibilità. Sono necessari alcuni componenti aggiuntivi per creare un'utile soluzione alta disponibilità.

- Riconnessione client e canale per il trasferimento di connessioni IBM MQ al computer che assume il controllo dell'esecuzione dell'istanza del gestore code attiva.
- Un file system di rete condiviso ad alte prestazioni (NFS) che gestisce correttamente i blocchi e fornisce la protezione contro gli errori del supporto e del server di file.

Importante: È necessario arrestare tutte le istanze del gestore code a più istanze in esecuzione nel proprio ambiente prima di eseguire la manutenzione sull'unità NFS . Verificare di disporre di backup di configurazione del gestore code da ripristinare, in caso di errore NFS .

- Reti e alimentatori resilienti per eliminare i singoli punti di errore nell'infrastruttura base.
- Applicazioni che tollerano il failover. In particolare, è necessario prestare particolare attenzione al comportamento delle applicazioni transazionali e alle applicazioni che sfogliano le code IBM MQ .
- Monitoraggio e gestione delle istanze attive e standby per garantire che siano in esecuzione e per riavviare le istanze attive che hanno avuto esito negativo. Sebbene i gestori code a più istanze si riavvino automaticamente, è necessario essere certi che le istanze in standby siano in esecuzione, pronte a prendere il controllo e che le istanze in errore vengano riportate in linea come nuove istanze in standby.

IBM MQ MQI clients e i canali si riconnettono automaticamente al gestore code in standby quando diventa attivo. Ulteriori informazioni sulla riconnessione e sugli altri componenti in una soluzione alta disponibilità sono disponibili negli argomenti correlati. La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

Piattaforme supportate

È possibile creare un gestore code a più istanze su qualsiasi sistema Multiplatforms.

La riconnessione automatica del client è supportata per client MQI.

Crea un gestore code a più istanze

Creare un gestore code a più istanze, creare il gestore code su un server e configurare IBM MQ su un altro server. I gestori code a più istanze condividono dati e log del gestore code.

La maggior parte delle attività coinvolte nella creazione di un gestore code a più istanze è l'attività di impostazione dei file di log e dei dati del gestore code condivisi. È necessario creare directory condivise sulla memoria di rete e rendere le directory disponibili per altri server utilizzando le condivisioni di rete. Queste attività devono essere eseguite da un utente con autorità amministrativa, ad esempio *root* su sistemi AIX and Linux . Le operazioni da eseguire vengono riportate di seguito.

1. Creare le condivisioni per i file di dati e di log.
2. Creare il gestore code su un server.
3. Eseguire il comando **dspmqlnf** sul primo server per raccogliere i dati di configurazione del gestore code e copiarli negli appunti.
4. Eseguire il comando **addmqinf** con i dati copiati per creare la configurazione del gestore code sul secondo server.

crtmqm non viene eseguito per creare nuovamente il gestore code sul secondo server.

Controllo accesso file

È necessario fare attenzione che l'utente e il gruppo *mqm* su tutti gli altri server abbiano l'autorizzazione per accedere alle condivisioni.

Su AIX and Linux, è necessario rendere *uid* e *gid* di *mqm* uguali su tutti i sistemi. Potrebbe essere necessario modificare */etc/passwd* su ciascun sistema per impostare un *uid* e *gid* comune per *mqm*, quindi riavviare il sistema.

Su Microsoft Windows, l'ID utente che sta eseguendo i processi del gestore code deve disporre dell'autorizzazione di controllo completo per le directory contenenti i file di log e i dati del gestore code. È possibile configurare l'autorizzazione in due modi:

1. Creare un gestore code con un gruppo globale come principal di sicurezza alternativo. Autorizzare il gruppo globale ad avere il controllo completo delle directory contenenti i dati del gestore code e i file di log; consultare [“Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows”](#) a pagina 554. Rendere l'ID utente che sta eseguendo il gestore code un membro del gruppo globale. Non è possibile rendere un utente locale membro di un gruppo globale, pertanto i processi del gestore code devono essere eseguiti con un ID utente del dominio. L'ID utente dominio deve essere un membro del gruppo locale mqm. L'attività, [“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows”](#) a pagina 529, illustra come impostare un gestore code a più istanze utilizzando i file protetti in questo modo.
2. Creare un gestore code sul controller di dominio, in modo che il gruppo mqm locale abbia un ambito dominio, "dominio locale". Proteggere la condivisione file con il dominio locale mqmed eseguire i processi del gestore code su tutte le istanze di un gestore code nello stesso gruppo mqm locale del dominio. L'attività, [“Creazione di un gestore code a più istanze su controller di dominio Windows”](#) a pagina 544, illustra come impostare un gestore code a più istanze utilizzando i file protetti in questo modo.

Informazioni sulla configurazione

Configurare tutte le istanze del gestore code necessarie modificando le informazioni di configurazione del gestore code IBM MQ per ciascun server. Ogni server deve avere la stessa versione di IBM MQ installata a un livello di correzioni compatibile. I comandi, **dspmqlnf** e **addmqinf**, consentono di configurare le istanze aggiuntive del gestore code. In alternativa, è possibile modificare direttamente i file `mqs.ini` e `qm.ini`. Gli argomenti, [“Creazione di un gestore code a più istanze su Linux”](#) a pagina 566, [“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows”](#) a pagina 529 e [“Creazione di un gestore code a più istanze su controller di dominio Windows”](#) a pagina 544 sono esempi che mostrano come configurare un gestore code a più istanze.


Su sistemi AIX, Linux, and Windows, è possibile condividere un singolo file `mqs.ini` posizionandolo sulla condivisione di rete e impostando la variabile di ambiente **AMQ_MQS_INI_LOCATION** in modo che punti ad esso.

Limitazioni

1. Configurare più istanze dello stesso gestore code solo su server con lo stesso sistema operativo, architettura ed endianness. Ad esempio, entrambe le macchine devono essere a 32 bit o a 64 bit.
2. Tutte le installazioni IBM MQ devono essere al livello di rilascio 7.0.1 o superiore.
3. Generalmente, le installazioni attive e in standby vengono mantenute allo stesso livello di manutenzione. Consultare le istruzioni di manutenzione per ogni aggiornamento per verificare se è necessario aggiornare tutte le installazioni insieme.

Tenere presente che i livelli di manutenzione per i gestori code attivi e passivi devono essere identici.

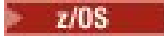
4. Condividere i dati e i log del gestore code solo tra gestori code configurati con lo stesso meccanismo di controllo accessi, gruppo e utente IBM MQ.  Ad esempio, la condivisione di rete configurata su un server Linux potrebbe contenere log e dati del gestore code separati per gestori code AIX and Linux, ma non potrebbe contenere i dati del gestore code utilizzati da IBM i.

 È possibile creare più condivisioni sulla stessa memoria di rete per sistemi IBM i e AIX and Linux purché le condivisioni siano diverse. È possibile assegnare diverse condivisioni a diversi proprietari. La limitazione è una conseguenza dei diversi nomi utilizzati per utenti e gruppi IBM MQ tra AIX and Linux e IBM i. Il fatto che l'utente e il gruppo possono avere gli stessi `uid` e `gid` non allentano la limitazione.

5. Su sistemi AIX and Linux, configurare il filesystem condiviso sull'archiviazione in rete con un montaggio `hard`, interrompibile, anziché un montaggio `soft`. Un montaggio `hard` interruptible forza il

blocco del gestore code fino a quando non viene interrotto da una chiamata di sistema. I montaggi soft non garantiscono la coerenza dei dati dopo un malfunzionamento del server.

6. Le directory di dati e di log condivisi non possono essere memorizzate su un file system FAT o NFSv3 . Per i gestori code a più istanze su Windows, la memoria di rete deve essere accessibile dal protocollo CIFS (Common Internet File System) utilizzato dalle reti Windows .

7.  z/OS non supporta gestori code a più istanze. Utilizzare i gruppi di condivisione code.

I client ricollegabili funzionano con i gestori code z/OS .

Domini Windows e gestori code a più istanze

Un gestore code a più istanze su Windows richiede che i relativi dati e log siano condivisi. La condivisione deve essere accessibile a tutte le istanze del gestore code in esecuzione su server o workstation differenti. Configurare i gestori code e condividere come parte di un dominio Windows . Il gestore code può essere eseguito su una stazione di lavoro o su un server di dominio o sul controller di dominio.

Importante: Per impostazione predefinita, i computer che iniziano con Windows 10 versione 1607 e Windows Server 2016 sono più restrittivi rispetto alle versioni precedenti di Windows.

Questa modifica limita i client a cui è consentito effettuare chiamate remote a SAM (Security Accounts Manager) e potrebbe influire su IBM MQ con l'avvio non riuscito dei gestori code. L'accesso a SAM è fondamentale per il funzionamento di IBM MQ quando IBM MQ viene configurato come account di dominio.

Prima di configurare un gestore code a più istanze, leggere [“Proteggere i file e le directory di log e i dati del gestore code non condivisi su Windows”](#) a pagina 557 e [“Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows”](#) a pagina 554 per esaminare come controllare l'accesso a dati del gestore code e file di log. Gli argomenti sono didattici; se si desidera passare direttamente alla configurazione delle directory condivise per un gestore code a più istanze in un dominio Windows , consultare [“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows”](#) a pagina 529.

Eseguire un gestore code a più istanze su server o workstation di dominio

Da IBM WebSphere MQ 7.1, i gestori code a più istanze vengono eseguiti su una workstation o su un server membro di un dominio. Per eseguire un gestore code a più istanze su Windows, è necessario un controller di dominio, un server di file e due stazioni di lavoro o server che eseguono lo stesso gestore code connesso allo stesso dominio.

La modifica che rende possibile l'esecuzione di un gestore code a più istanze su qualsiasi server o stazione di lavoro in un dominio, è che ora è possibile creare un gestore code con un gruppo di sicurezza aggiuntivo. Il gruppo di sicurezza aggiuntivo viene passato nel parametro -a del comando `crtmqm` . Proteggere le directory che contengono i dati e i log del gestore code con il gruppo. L'ID utente che esegue i processi del gestore code deve essere un membro di questo gruppo. Quando il gestore code accede alle directory, Windows controlla le autorizzazioni di cui l'ID utente dispone per accedere alle directory. Fornendo sia il gruppo che l'ambito di dominio dell'ID utente, l'ID utente che esegue i processi del gestore code ha le credenziali del gruppo globale. Quando il gestore code è in esecuzione su un altro server, l'ID dell'utente che esegue i processi del gestore code può avere le stesse credenziali. L'ID utente non deve essere lo stesso. Deve essere un membro del gruppo di sicurezza alternativo, nonché un membro del gruppo `mqm` locale.

Consultare [“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows”](#) a pagina 529 per i dettagli sulla creazione di un gestore code a più istanze.

Sono richiesti più passi per configurare il dominio, i server di dominio e le stazioni di lavoro. È necessario comprendere il modo in cui Windows autorizza l'accesso da parte di un gestore code alle relative directory di dati e log. Se non si è certi di come i processi del gestore code siano autorizzati ad accedere ai relativi file di log e di dati, leggere l'argomento [“Proteggere i file e le directory di log e i dati del gestore code non condivisi su Windows”](#) a pagina 557. L'argomento include due attività che consentono di comprendere i passaggi richiesti. Le attività sono [“Lettura e scrittura di dati e file di log autorizzati dal gruppo `mqm` locale”](#) a pagina 559 e [“Lettura e scrittura dei dati e dei file di log autorizzati da un gruppo di sicurezza](#)

locale alternativo” a pagina 562. Un altro argomento, “Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows” a pagina 554, spiega come proteggere le directory condivise contenenti i dati del gestore code e i file di log con il gruppo di protezione alternativo. L'argomento include quattro attività, per configurare un dominio Windows , creare una condivisione file, installare IBM MQ for Windows configurare un gestore code per utilizzare la condivisione. Le attività sono le seguenti:

1. [“Creazione di un dominio Active Directory e DNS su Windows” a pagina 532.](#)
2. [“Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows” a pagina 536.](#)
3. [“Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows” a pagina 539.](#)
4. [“Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo” a pagina 541.](#)

È quindi possibile eseguire l'attività, [“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows” a pagina 529](#), utilizzando il dominio. Eseguire queste operazioni per esplorare la configurazione di un gestore code a più istanze prima di trasferire le proprie conoscenze a un dominio di produzione.

Eseguire un gestore code a più istanze sui controller di dominio

I dati del gestore code potrebbero essere protetti con il gruppo mqm del dominio. Come spiegato nell'argomento [“Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows” a pagina 554](#) , non è possibile condividere le directory protette con il gruppo mqm locale su workstation o server. Tuttavia, sui controller di dominio, tutti i gruppi e i principal hanno un ambito di dominio. Se si installa IBM MQ for Windows su un controller di dominio, i dati del gestore code e i file di log sono protetti con il gruppo mqm del dominio, che può essere condiviso. Attenersi alla procedura riportata nell'attività, [“Creazione di un gestore code a più istanze su controller di dominio Windows” a pagina 544](#) per configurare un gestore code a più istanze sui controller di dominio.

Informazioni correlate

[Gestione dell'autorizzazione e del controllo degli accessi](#)

[Come utilizzare i nodi cluster di Windows Server come controller di dominio](#)

Windows *Creazione di un gestore code a più istanze su workstation o server di dominio su Windows*

Un esempio mostra come impostare un gestore code a più istanze su Windows su una stazione di lavoro o su un server che fa parte di un dominio Windows . Il server non deve essere un controller di dominio. La configurazione dimostra i concetti coinvolti, piuttosto che essere una scala di produzione. L'esempio è basato su Windows Server 2008. La procedura potrebbe differire su altre versioni di Windows Server.

In una configurazione della scala di produzione, potrebbe essere necessario adattare la configurazione a un dominio esistente. Ad esempio, è possibile definire diversi gruppi di domini per autorizzare diverse condivisioni e per raggruppare gli ID utente che eseguono gestori code.

La configurazione di esempio è composta da tre server:

sun

Un controller di dominio Windows Server 2008. Possiede il dominio *wmq.example.com* che contiene *Sun, marse venus*. A scopo illustrativo, viene utilizzato anche come server di file.

mars

Un Windows Server 2008 utilizzato come primo server IBM MQ . Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

venus

Un Windows Server 2008 utilizzato come secondo server IBM MQ . Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

Su Windows, non è necessario verificare il file system su cui si intende memorizzare i dati del gestore code e i file di log. La procedura di controllo, [Verifica del funzionamento del file system condiviso](#), è applicabile a AIX and Linux. Su Windows, le verifiche hanno sempre esito positivo.

Effettuare le operazioni riportate di seguito. Le attività creano il controller di dominio e il dominio, installano IBM MQ for Windows su un server e creano la condivisione file per i file di dati e di log. Se si sta configurando un controller di dominio esistente, potrebbe essere utile provare la procedura su un nuovo server Windows 2008. È possibile adattare la procedura al dominio.

1. [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532.
2. [“Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows”](#) a pagina 536.
3. [“Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows”](#) a pagina 539.
4. [“Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo”](#) a pagina 541.

Informazioni su questa attività

Questa attività fa parte di una sequenza di attività per configurare un controller di dominio e due server nel dominio per eseguire le istanze di un gestore code. In questa attività è possibile configurare un secondo server, *venus*, per eseguire un'altra istanza del gestore code *QMGR*. Seguire i passi in questa attività per creare la seconda istanza del gestore code, *QMGR*, e verificare che funzioni.

Questa attività è separata dalle quattro attività della precedente sezione. Contiene i passi che convertono un gestore code a istanza singola in un gestore code a più istanze. Tutti gli altri passi sono comuni ai gestori code a istanza singola o multipla.

Procedura

1. Configurare un secondo server per eseguire IBM MQ for Windows.
 - a) Effettuare le operazioni riportate nell'attività [“Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows”](#) a pagina 536 per creare un secondo server di dominio. In questa sequenza di attività il secondo server è denominato *venus*.
Suggerimento: Creare la seconda installazione utilizzando gli stessi valori predefiniti di installazione per IBM MQ su ognuno dei due server. Se i valori predefiniti sono differenti, potrebbe essere necessario adattare le variabili *Prefisso* e *InstallationName* nella sezione *QMGR QueueManager* nel IBM MQ file di configurazione *mqs.ini*. Le variabili fanno riferimento a percorsi che possono essere diversi per ogni installazione e gestore code su ciascun server. Se i percorsi rimangono gli stessi su ogni server, è più semplice configurare un gestore code a più istanze.
2. Creare una seconda istanza di *QMGR* su *venus*.
 - a) Se *QMGR* su *mars* non esiste, eseguire l'attività [“Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo”](#) a pagina 541 per crearla
 - b) Controllare che i parametri *Prefisso* e *InstallationName* siano corretti per *venus*.

In *mars*, eseguire il comando **dspmqinf** :

```
dspmqinf QMGR
```

La risposta del sistema:

```
QueueManager:  
Name=QMGR
```

```
Directory=QMGR
Prefix=C:\ProgramData\IBM\MQ
DataPath=\\sun\wmq\data\QMGR
InstallationName=Installation1
```

- c) Copiare il formato leggibile dalla macchina della stanza **QueueManager** negli appunti.
Su *mars* , eseguire nuovamente il comando **dspmqrinf** con il parametro -o command .

```
dspmqrinf -o command QMGR
```

La risposta del sistema:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

- d) In *venus* eseguire il comando **addmqinf** dagli appunti per creare un'istanza del gestore code su *venus*.

Modificare il comando, se necessario, per adattare le differenze nei parametri Prefisso o InstallationName .

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

IBM MQ configuration information added.

3. Avviare il gestore code *QMGR* su *venus*, consentendo le istanze in standby.

- a) Verificare che *QMGR* on *mars* sia arrestato.

In *mars*, eseguire il comando **dspmqr** :

```
dspmqr -m QMGR
```

La risposta del sistema dipende da come è stato arrestato il gestore code; ad esempio:

```
C:\Users\Administrator>dspmqr -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

- b) Su *venus* eseguire il comando **strmqm** per avviare *QMGR* consentendo gli standby:

```
strmqm -x QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

Risultati

Per verificare gli switch del gestore code a più istanze, effettuare le seguenti operazioni:

1. Su *mars*, eseguire il comando **strmqm** per avviare *QMGR* consentendo standby:

```
strmqm -x QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
A standby instance of queue manager 'QMGR' has been started.  
The active instance is running elsewhere.
```

2. Su *venus* eseguire il comando **endmqm** :

```
endmqm -r -s -i QMGR
```

La risposta del sistema su *venus*:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended, permitting switchover to  
a standby instance.
```

E su *mars*:

```
dspmq  
QMNAME(QMGR) STATUS(Running as standby)  
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running as standby)  
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running)
```

Operazioni successive

Per verificare un gestore code a più istanze utilizzando programmi di esempio, consultare [“Verifica del gestore code a più istanze su Windows”](#) a pagina 552.

Windows Creazione di un dominio Active Directory e DNS su Windows

Questa attività crea il dominio *wmq.example.com* su un controller di dominio Windows 2008 denominato *sun*. Configura il gruppo globale Domain *mqm* nel dominio, con i diritti corretti e con un utente.

In una configurazione della scala di produzione, potrebbe essere necessario adattare la configurazione a un dominio esistente. Ad esempio, è possibile definire diversi gruppi di domini per autorizzare diverse condivisioni e per raggruppare gli ID utente che eseguono gestori code.

La configurazione di esempio è composta da tre server:

sun

Un controller di dominio Windows Server 2008. Possiede il dominio *wmq.example.com* che contiene *Sun*, *marse* *venus*. A scopo illustrativo, viene utilizzato anche come server di file.

mars

Un Windows Server 2008 utilizzato come primo server IBM MQ . Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

venus

Un Windows Server 2008 utilizzato come secondo server IBM MQ . Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

1. I passi dell'attività sono congruenti con Windows Server 2008 installato ma non configurato con alcun ruolo. Se si sta configurando un controller di dominio esistente, potrebbe essere utile provare la procedura su un nuovo server Windows 2008. È possibile adattare la procedura al dominio.

Informazioni su questa attività

In questa attività, si crea un dominio Active Directory e DNS su un nuovo controller di dominio. È quindi possibile configurarlo per installare IBM MQ su altri server e stazioni di lavoro che si uniscono al dominio. Seguire l'attività se non si ha familiarità con l'installazione e la configurazione di Active Directory per creare un dominio Windows . È necessario creare un dominio Windows per creare una configurazione del gestore code a più istanze. L'attività non ha lo scopo di guidare l'utente nel modo migliore per configurare un dominio Windows . Per distribuire gestori code a più istanze in un ambiente di produzione, è necessario consultare la documentazione Windows .

Durante l'attività, effettuare le seguenti operazioni:

1. Installare Active Directory.
2. Aggiunge un dominio.
3. Aggiungere il dominio a DNS.
4. Creare il gruppo globale Domain *mqm* e assegnarle i diritti corretti.
5. Aggiungere un utente e renderlo membro del gruppo globale Domain *mqm*.

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze”](#) a pagina 528.

Per l'attività il nome host del controller di dominio è *sune* i due server IBM MQ sono denominati *mars* e *venus*. Il dominio è denominato *wmq.example.com*. È possibile sostituire tutti i nomi in corsivo nell'attività con nomi di propria scelta.

Procedura

1. Accedere al controller di dominio, *sun*, come amministratore locale o Workgroup .
Se il server è già configurato come controller di dominio, è necessario collegarsi come amministratore di dominio.
2. Eseguire la procedura guidata Servizi dominio di Active Directory .
 - a) Fare clic su **Avvia > Esegui ...** Immettere *dcpromo* e fare clic su **OK**.
Se i file binari di Active Directory non sono già installati, Windows installa automaticamente i file.
3. Nella prima finestra della procedura guidata, lasciare deselezionata la check box **Utilizza installazione in modalità avanzata** . Fare clic su **Avanti > Avanti** e selezionare **Crea un nuovo dominio in un nuovo insieme di strutture > Avanti**.
4. Immettere *wmq.example.com* nel campo **FQDN del dominio root dell'insieme di strutture** . Fare clic su **Avanti**.
5. Nella finestra Imposta livello funzionale dell'insieme di strutture, selezionare **Windows Server 2003**, o successivo, dall'elenco di **Livelli funzionali dell'insieme di strutture > Avanti**.
Il livello più vecchio di Windows Server supportato da IBM MQ è Windows Server 2003.

6. Opzionale: Nella finestra Imposta livello funzionale del dominio, selezionare **Windows Server 2003** o successivo dall'elenco di **Livelli funzionali del dominio > Avanti**.
Questo passo è richiesto solo se si imposta il livello funzionale dell'insieme di strutture su **Windows server 2003**.
7. Viene visualizzata la finestra Opzioni controller di dominio aggiuntive, con **Server DNS** selezionato come opzione aggiuntiva. Fare clic su **Avanti** e su **Sì** per cancellare la finestra di avvertenza.
Suggerimento: Se un server DNS è già installato, questa opzione non viene visualizzata. Se si desidera seguire questa attività in modo preciso, rimuovere tutti i ruoli da questo controller di dominio e riavviare.
8. Lasciare invariate le directory Database, Log File e SYSVOL ; fare clic su **Avanti**.
9. Immettere una parola d'ordine nei campi **Password** e **Conferma parola d'ordine** nella finestra Directory Services Restore Mode Administrator Password. Fare clic su **Avanti > Avanti**. Selezionare **Riavvia al completamento** nella finestra finale della procedura guidata.
10. Quando il controller di dominio viene riavviato, collegarsi come `wmq\Administrator`.
Il gestore server viene avviato automaticamente.
11. Aprire la cartella `wmq.example.com\Users`
 - a) Aprire **Server Manager > Ruoli > Active Directory Domain Services > wmq.example.com > Utenti**.
12. Fare clic con il tasto destro del mouse su **Utenti > Nuovo gruppo > .**
 - a) Immettere un nome gruppo nel campo **Nome gruppo**.
Nota: Il nome gruppo preferito è `Domain mqm`. Immetterlo esattamente come visualizzato.
 - Richiamando il gruppo `Domain mqm` si modifica il comportamento di Prepare IBM MQ Wizard su una stazione di lavoro o su un server di dominio. Fa sì che Prepare IBM MQ Wizard aggiunga automaticamente il gruppo `Domain mqm` al gruppo `mqm` locale ad ogni nuova installazione di IBM MQ nel dominio.
 - È possibile installare workstation o server in un dominio senza alcun gruppo globale `Domain mqm`. In questo caso, è necessario definire un gruppo con le stesse proprietà del gruppo `Domain mqm`. È necessario rendere il gruppo o gli utenti membri di esso, dei membri del gruppo `mqm` locale ovunque IBM MQ sia installato in un dominio. È possibile inserire gli utenti di dominio in più gruppi. Creare più gruppi di domini, ciascuno dei quali corrispondente a un insieme di installazioni che si desidera gestire separatamente. Assegnare gli utenti di dominio ai diversi gruppi di domini in base alle installazioni gestite. Aggiungere ciascun gruppo o gruppi di domini al gruppo `mqm` locale delle diverse installazioni di IBM MQ. Solo gli utenti di dominio nei gruppi di domini che sono dei membri di un gruppo `mqm` locale specifico possono creare, gestire ed eseguire i gestori code per tale installazione.
 - L'utente del dominio che si nomina quando si installa IBM MQ su una workstation o su un server in un dominio deve essere un membro del gruppo `Domain mqm` o di un gruppo alternativo definito con le stesse proprietà del gruppo `Domain mqm`.
 - b) Lasciare selezionato **Globale** come **Ambito del gruppo** o modificarlo in **Universale**. Lasciare selezionato **Sicurezza** come **Tipo di gruppo**. Fare clic su **OK**.
13. Aggiungere i diritti, **Consenti Lettura appartenenza gruppo** e **Consenti Leggi groupMembershipSAM** ai diritti del gruppo globale `Domain mqm`.
 - a) Nella barra delle azioni di Server Manager, fare clic su **Visualizza > Funzioni avanzate**
 - b) Nella struttura ad albero di navigazione di Server Manager, fare clic su **Utenti**
 - c) Nella finestra Utenti, fare clic con il pulsante destro del mouse su **Domain mqm > Proprietà**
 - d) Fare clic su **Sicurezza > Avanzate > Aggiungi** Immettere `Domain mqm` e fare clic su **Controlla nomi > OK**.
Il campo **Nome** è precompilato con la stringa `Domain mqm (domain name\Domain mqm)`.
 - e) Fare clic su **Proprietà**. Nell'elenco **Applica a**, selezionare **Oggetti utente discendenti**.

- f) Dall'elenco **Autorizzazioni** , selezionare le caselle di spunta **Leggi appartenenza gruppo** e **Leggi groupMembershipSAM Consenti** ; fare clic su **OK > Applica > OK > OK**.

14. Aggiungere due o più utenti al gruppo globale Domain mqm .

Un utente, *wmquser1* nell'esempio, esegue il servizio IBM MQ e l'altro utente, *wmquser2* , viene utilizzato in modo interattivo.

Un utente di dominio è richiesto per creare un gestore code che utilizza il gruppo di sicurezza alternativo in una configurazione di dominio. Non è sufficiente che l'ID utente sia un amministratore, anche se un amministratore dispone dell'autorizzazione per eseguire il comando **crtmqm** . L'utente del dominio, che potrebbe essere un amministratore, deve appartenere al gruppo mqm locale e al gruppo di protezione alternativo.

Nell'esempio, si rendono membri *wmquser1* e *wmquser2* del gruppo globale Domain mqm . Prepare IBM MQ Wizard configura automaticamente Domain mqm come membro del gruppo mqm locale in cui viene eseguita la procedura guidata.

È necessario fornire un altro utente per eseguire il servizio IBM MQ per ciascuna installazione di IBM MQ su un singolo computer. È possibile riutilizzare gli stessi utenti su computer differenti.

- Nella struttura ad albero di navigazione Server Manager, fare clic su **Utenti > Nuovo > Utente**
- Nella finestra Nuovo oggetto - Utente, immettere *wmquser1* nel campo **Nome di accesso utente** . Immettere *WebSphere* nel campo **Nome** e *MQ1* nel campo **Cognome** . Fare clic su **Avanti**.
- Immettere una password nei campi **Password** e **Conferma password** e deselezionare la casella di spunta **L'utente deve modificare la password al successivo accesso** . Fare clic su **Avanti > Fine**.
- Nella finestra Utenti, fare clic con il tasto destro del mouse su **WebSphere MQ > Aggiungi a un gruppo ...** . Digitare Domain mqm e fare clic su **Verifica nomi > OK > OK**.
- Ripetere i passi da **a** a **d** per aggiungere *WebSphere MQ2* come *wmquser2* .

15. Esecuzione di IBM MQ come servizio.

Se è necessario eseguire IBM MQ come un servizio e fornire all'utente del dominio (ottenuto dall'amministratore del dominio) l'accesso per l'esecuzione come un servizio, attenersi alla seguente procedura:

- Fare clic su **Start > Esegui ...**
Immettere il comando *secpo1.msc* e fare clic su **OK**.
- Aprire **Impostazioni di sicurezza > Politiche locali > Assegnazioni diritti utente**.
Nell'elenco delle politiche, fare clic con il pulsante destro del mouse su **Accedi come servizio > Proprietà**.
- Fare clic su **Aggiungi utente o gruppo ...**
Immettere il nome dell'utente ottenuto dall'amministratore del dominio e fare clic su **Verifica nomi**
- Se richiesto da una finestra Sicurezza Windows , immettere il nome utente e la password di un utente o di un responsabile dell'account con autorizzazione sufficiente e fare clic su **OK > Applica > OK**.
Chiudere la finestra Politica di sicurezza locale.

Nota: Su Windows Server 2008 e Windows Server 2012, l'UAC (User Account Control) è abilitato per impostazione predefinita.

La funzione UAC limita le azioni che gli utenti possono eseguire su alcune funzioni del sistema operativo, anche se sono dei membri del gruppo di amministratori. È necessario prendere le misure appropriate per risolvere questa limitazione.

Operazioni successive

Procedere con l'attività successiva, "Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows" a pagina 536.

Attività correlate

Windows [Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows](#)

Windows [Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows](#)

Windows [Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo](#)

Windows [Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows](#)

In questa attività, si installerà e si configurerà IBM MQ su un server o su una stazione di lavoro nel dominio *wmq.example.com* Windows .

In una configurazione della scala di produzione, potrebbe essere necessario adattare la configurazione a un dominio esistente. Ad esempio, è possibile definire diversi gruppi di domini per autorizzare diverse condivisioni e per raggruppare gli ID utente che eseguono gestori code.

La configurazione di esempio è composta da tre server:

sun

Un controller di dominio Windows Server 2008. Possiede il dominio *wmq.example.com* che contiene *Sun, marse venus*. A scopo illustrativo, viene utilizzato anche come server di file.

mars

Un Windows Server 2008 utilizzato come primo server IBM MQ . Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

venus

Un Windows Server 2008 utilizzato come secondo server IBM MQ . Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

Importante: Per impostazione predefinita, i computer che iniziano con Windows 10 versione 1607 e Windows Server 2016 sono più restrittivi rispetto alle versioni precedenti di Windows.

Questa modifica limita i client a cui è consentito effettuare chiamate remote a SAM (Security Accounts Manager) e potrebbe influire su IBM MQ con l'avvio non riuscito dei gestori code. L'accesso a SAM è fondamentale per il funzionamento di IBM MQ quando IBM MQ viene configurato come account di dominio.

1. Effettuare le operazioni in [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532 per creare un controller di dominio, *sun*, per il dominio *wmq.example.com*. Modificare i nomi in corsivo per adattarli alla propria configurazione.
2. Consultare [Requisiti hardware e software sui sistemi Windows](#) per altre versioni Windows su cui è possibile eseguire IBM MQ .

Informazioni su questa attività

In questa attività è possibile configurare un server Windows 2008, denominato *mars*, come membro del dominio *wmq.example.com* . Si installa IBM MQe si configura l'installazione da eseguire come membro del dominio *wmq.example.com* .

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze”](#) a pagina 528.

Per l'attività il nome host del controller di dominio è *sune* i due server IBM MQ sono denominati *mars* e *venus*. Il dominio è denominato *wmq.example.com*. È possibile sostituire tutti i nomi in corsivo nell'attività con nomi di propria scelta.

Procedura

1. Aggiungere il controller di dominio, *sun.wmq.example.com* a *mars* come server DNS.
 - a) Su *mars*, collegarsi come *mars\Administrator* e fare clic su **Avvia**.
 - b) Fare clic con il pulsante destro del mouse su **Rete > Proprietà > Gestisci connessioni di rete**.
 - c) Fare clic con il tasto destro del mouse sull'adattatore di rete, fare clic su **Proprietà**.

Il sistema risponde con la finestra Proprietà connessione area locale che elenca le voci utilizzate dalla connessione.
 - d) Selezionare **Internet Protocol Versione 4** o **Internet Protocol IBM WebSphere MQ 6** dall'elenco di voci nella finestra Proprietà di connessione alla rete locale. Fare clic su **Proprietà > Avanzate ...** e fare clic sul separatore **DNS**.
 - e) Sotto gli indirizzi server DNS, fare clic su **Aggiungi ...**.
 - f) Immettere l'indirizzo IP del controller di dominio, che è anche il server DNS, e fare clic su **Aggiungi**.
 - g) Fare clic su **Aggiungi questi suffissi DNS > Aggiungi**
 - h) Immettere *wmq.example.com* e fare clic su **Aggiungi**.
 - i) Immettere *wmq.example.com* nel campo **Suffisso DNS per questa connessione**.
 - j) Seleziona **Register this connection's address in DNS** e **Use this connection's suffix in DNS registration**. Fare clic su **OK > OK > Chiudi**
 - k) Aprire una finestra comandi e immettere il comando **ipconfig /all** per esaminare le impostazioni TCP/IP.
2. Su *mars*, aggiungere il computer al dominio *wmq.example.com*.
 - a) Fare clic su **Avvia**
 - b) Fare clic con il tasto destro del mouse su **Computer > Proprietà**. Nella divisione Nome computer, dominio e impostazioni gruppo di lavoro, fare clic su **Modifica impostazioni**.
 - c) Nelle finestre delle proprietà del sistema, fare clic su **Modifica**.
 - d) Fare clic su Dominio, immettere *wmq.example.com* e fare clic su **OK**.
 - e) Immettere il **Nome utente** e la **Password** dell'amministratore del controller di dominio, che dispone dell'autorizzazione per consentire al computer di unirsi al dominio e fare clic su **OK**.
 - f) Fare clic su **OK > OK > Chiudi > Riavvia ora** in risposta al messaggio "Benvenuti nel dominio *wmq.example.com*".
3. Verificare che il computer sia un membro del dominio *wmq.example.com*
 - a) Su *sun*, accedere al controller di dominio come *wmq\Administrator*.
 - b) Aprire **Server Manager > Active Directory Domain Services > wmq.example.com > Computer** e controllare che *mars* sia elencato correttamente nella finestra Computer.
4. Installare IBM MQ for Windows su *mars*.

Per ulteriori informazioni sull'esecuzione della procedura guidata di installazione di IBM MQ for Windows, consultare [Installazione del server IBM MQ su Windows](#).

 - a) Su *mars*, accedere come amministratore locale, *mars\Administrator*.
 - b) Eseguire il comando **Setup** sul supporto di installazione IBM MQ for Windows.

Viene avviata l'applicazione IBM MQ Launchpad.
 - c) Fare clic su **Requisiti software** per verificare che il software prerequisito sia installato.
 - d) Fare clic su **Configurazione di rete > Sì** per configurare un ID utente di dominio.

L'attività, "Creazione di un dominio Active Directory e DNS su Windows" a pagina 532, configura un ID utente di dominio per questa serie di attività.
 - e) Fare clic su **IBM MQ Installazione**, selezionare una lingua di installazione e fare clic su **Avvia IBM MQ Installer**.

f) Confermare l'accordo di licenza e fare clic su **Avanti > Avanti > Installa** per accettare la configurazione predefinita. Attendere il completamento dell'installazione e fare clic su **Fine**.

È possibile modificare il nome dell'installazione, installare componenti differenti, configurare una directory differente per i dati e i log del gestore code o installare in una directory differente. In tal caso, fare clic su **Personalizzato** invece di **Tipico**.

IBM MQ è installato e il programma di installazione avvia Prepare IBM MQ Wizard.

Importante: Non eseguire ancora la procedura guidata.

5. Configurare l'utente che eseguirà il servizio IBM MQ con il diritto **Esegui come servizio**.

Scegliere se configurare il gruppo mqm locale, il gruppo Domain mqm o l'utente che eseguirà il servizio IBM MQ con la destra. Nell'esempio, si dà all'utente il diritto.

- Fare clic su **Avvia > Esegui ...**, immettere il comando **secpol.msc** e fare clic su **OK**.
- Aprire **Impostazioni di protezione > Politiche locali > Assegnazione diritti utente**. Nell'elenco delle politiche, fare clic con il tasto destro del mouse su **Accedi come servizio > Proprietà**.
- Fare clic su **Aggiungi utente o gruppo ...** e immettere *wmquser1* e fare clic su **Verifica nomi**
- Immettere il nome utente e la password di un amministratore di dominio, *wmq\Administratore* fare clic su **OK > Applica > OK**. Chiudere la finestra Politica di sicurezza locale.

6. Eseguire Prepare IBM MQ Wizard.

Per ulteriori informazioni, consultare [Configurazione di IBM MQ con Prepare IBM MQ Wizard](#).

a) Il programma di installazione IBM MQ esegue automaticamente Prepare IBM MQ Wizard.

Per avviare manualmente la procedura guidata, individuare il collegamento alla cartella Prepare IBM MQ Wizard nella cartella **Start > Tutti i programmi > IBM MQ**. Selezionare il collegamento che corrisponde all'installazione di IBM MQ in una configurazione a più installazioni.

- Fare clic su **Avanti** e lasciare **Si** selezionato in risposta alla domanda "Identifica se è presente un controller di dominio Windows 2000 o successivo nella rete".
- Fare clic su **Si > Avanti** nella prima finestra Configurazione di IBM MQ for Windows per gli utenti del dominio Windows.
- Nella seconda finestra Configurazione di IBM MQ for Windows per gli utenti del dominio Windows, immettere *wmq* nel campo **Dominio**. Immettere *wmquser1* nel campo **Nome utente** e la password, se impostata, nel campo **Password**. Fare clic su **Avanti**.


La procedura guidata configura e avvia IBM MQ con *wmquser1*.


e) Nella pagina finale della procedura guidata, selezionare o deselezionare le caselle di spunta richieste e fare clic su **Fine**.


Operazioni successive

- Eseguire l'attività, ["Lettura e scrittura di dati e file di log autorizzati dal gruppo mqm locale"](#) a pagina 559, per verificare che l'installazione e la configurazione stiano funzionando correttamente.
- Eseguire l'attività, ["Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows"](#) a pagina 539, per impostare una condivisione file per memorizzare i dati e i file di log di un gestore code a più istanze.

Attività correlate

 [Creazione di un dominio Active Directory e DNS su Windows](#)

 [Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows](#)

 [Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo](#)

Riferimenti correlati

[Diritti utente richiesti per un servizio IBM MQ Windows](#)

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta.

In una configurazione della scala di produzione, potrebbe essere necessario adattare la configurazione a un dominio esistente. Ad esempio, è possibile definire diversi gruppi di domini per autorizzare diverse condivisioni e per raggruppare gli ID utente che eseguono gestori code.

La configurazione di esempio è composta da tre server:

sun

Un controller di dominio Windows Server 2008. Possiede il dominio *wmq.example.com* che contiene *Sun, marse venus*. A scopo illustrativo, viene utilizzato anche come server di file.

mars

Un Windows Server 2008 utilizzato come primo server IBM MQ . Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

venus

Un Windows Server 2008 utilizzato come secondo server IBM MQ . Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

1. Per eseguire questa attività esattamente come documentato, effettuare le operazioni riportate nell'attività, [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532, per creare il dominio *sun.wmq.example.com* sul controller di dominio *sun*. Modificare i nomi in corsivo per adattarli alla propria configurazione.

Informazioni su questa attività

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze”](#) a pagina 528.

Nell'attività, viene creata una condivisione contenente una directory di dati e log e un gruppo globale per autorizzare l'accesso alla condivisione. Si passa il nome del gruppo globale che autorizza la condivisione al comando **crtmqm** nel relativo parametro **-a** . Il gruppo globale offre la flessibilità di separare gli utenti di questa condivisione dagli utenti di altre azioni. Se non è necessaria questa flessibilità, autorizzare la condivisione con il gruppo `Domain mqm` piuttosto che creare un nuovo gruppo globale.

Il gruppo globale utilizzato per la condivisione in questa attività è denominato *wmqhae* la condivisione è denominata *wmq*. Sono definiti sul controller di dominio *sun* nel Windows dominio *wmq.example.com*. La condivisione ha autorizzazioni di controllo complete per il gruppo globale *wmqha*. Sostituire i nomi in corsivo nell'attività con i nomi desiderati.

Ai fini di questa attività, il controller di dominio è lo stesso server del server di file. Nelle applicazioni pratiche, suddividere i servizi di directory e file tra diversi server per prestazioni e disponibilità.

È necessario configurare l'ID utente con cui è in esecuzione il gestore code in modo che sia membro di due gruppi. Deve essere un membro del gruppo `mqm` locale su un server IBM MQ e del gruppo globale *wmqha* .

In questa serie di attività, quando il gestore code è in esecuzione come servizio, viene eseguito con l'ID utente *wmquser1*, quindi *wmquser1* deve essere un membro di *wmqha*. Quando il gestore code viene eseguito in modo interattivo, viene eseguito con l'ID utente *wmquser2*, quindi *wmquser2* deve essere un membro di *wmqha*. Sia *wmquser1* che *wmquser2* sono membri del gruppo globale `Domain mqm`. `Domain mqm` è un membro del gruppo `mqm` locale sui server *mars* e *venus* IBM MQ . Pertanto, *wmquser1* e *wmquser2* sono membri del gruppo `mqm` locale su entrambi i IBM MQ server.

Procedura

1. Accedere al controller di dominio, *sun.wmq.example.com* come amministratore del dominio.
2. Creare il gruppo globale *wmqha*.
 - a) Aprire **Server Manager** > **Ruoli** > **Active Directory Domain Services** > *wmq.example.com* > **Utenti**.
 - b) Aprire la cartella *wmq.example.com\Users*
 - c) Fare clic con il tasto destro del mouse su **Utenti** > **Nuovo gruppo** > .
 - d) Immettere *wmqha* nel campo **Nome gruppo** .
 - e) Lasciare **Globale** selezionato come **Ambito gruppo** e **Sicurezza** come **Tipo di gruppo**. Fare clic su **OK**.
3. Aggiungere gli utenti del dominio *wmquser1* e *wmquser2* al gruppo globale, *wmqha*.
 - a) Nella struttura ad albero di navigazione Server Manager, fare clic su **Utenti** e fare clic con il tasto destro del mouse su *wmqha* > **Proprietà** nell'elenco degli utenti.
 - b) Fare clic sulla scheda Membri nella finestra Proprietà di *wmqha* .
 - c) Fare clic su **Aggiungi ...** ; immettere *wmquser1* ; *wmquser2* e fare clic su **Verifica nomi** > **OK** > **Applica** > **OK**.
4. Creare la struttura di directory per contenere i dati del gestore code e i file di log.
 - a) Aprire un prompt dei comandi.
 - b) Immettere il comando:

```
md c:\wmq\data, c:\wmq\logs
```


5. Autorizzare il gruppo globale *wmqha* ad avere l'autorizzazione di controllo completo per le directory *c:\wmq* e la condivisione.
 - a) In Windows Explorer, fare clic con il pulsante destro del mouse su **c:\wmq** > **Proprietà**.
 - b) Fare clic sulla scheda **Sicurezza** , quindi su **Avanzate** > **Modifica**
 - c) Deselezionare la check box per **Includi autorizzazioni ereditabili dal proprietario di questo oggetto**. Fare clic su **Copia** nella finestra Sicurezza Windows .
 - d) Selezionare le linee per gli utenti nell'elenco **Voci di autorizzazione** e fare clic su **Rimuovi**. Lasciare le righe per SYSTEM, Administrators e CREATOR OWNER nell'elenco **Voci di autorizzazione**.
 - e) Fare clic su **Aggiungi ...**, e immettere il nome del gruppo globale *wmqha*. Fare clic su **Verifica nomi** > **OK**.
 - f) Nella finestra Immissione per *wmq* , selezionare **Controllo completo** nell'elenco di **Autorizzazioni**.
 - g) Fare clic su **OK** > **Applica** > **OK** > **OK** > **OK**
 - h) In Windows Explorer, fare clic con il pulsante destro del mouse su **c:\wmq** > **Condividi**
 - i) Fare clic su **Condivisione avanzata ...** e selezionare la casella di spunta **Condividi questa cartella** . Lasciare il nome condivisione come *wmq*.
 - j) Fare clic su **Autorizzazioni** > **Aggiungi ...**, e immettere il nome del gruppo globale *wmqha*. Fare clic su **Verifica nomi** > **OK**.
 - k) Selezionare *wmqha* nell'elenco **Nomi gruppo o utente**. Selezionare la casella di spunta **Controllo completo** nell'elenco **Autorizzazioni per wmqha** ; fare clic su **Applica**.
 - l) Selezionare *Administrators* nell'elenco **Nomi gruppo o utente**. Selezionare la casella di spunta **Controllo completo** nell'elenco **Autorizzazioni per gli amministratori** ; fare clic su **Applica** > **OK** > **OK** > **Chiudi**.


Operazioni successive


Verificare che sia possibile leggere e scrivere file nelle directory condivise da ciascuno dei server IBM MQ . Controllare l'ID utente di servizio IBM MQ *wmquser1* e l'ID utente interattivo, *wmquser2*.


1. Se si utilizza il desktop remoto, è necessario aggiungere *wmq\wmquser1* e *wmquser2* al gruppo locale Remote Desktop Users su *mars*.
 - a. Accedere a *mars* come *wmq\Administrator*
 - b. Eseguire il comando **lusrmgr.msc** per aprire la finestra Utenti e gruppi locali.
 - c. Fare clic su **Gruppi**. Fare clic con il pulsante destro del mouse su **Utenti desktop remoto > Proprietà > Aggiungi ...**. Immettere *wmquser1* ; *wmquser2* e fare clic su **Verifica nomi**.
 - d. Immettere il nome utente e la password dell'amministratore del dominio, *wmq\Administratore* fare clic su **OK > Applica > OK**.
 - e. Chiudere la finestra Utenti e gruppi locali.
2. Accedere a *mars* come *wmq\wmquser1*.
 - a. Aprire una finestra Esplora risorse di Windows e immettere `\\sun\wmq`.
Il sistema risponde aprendo la condivisione *wmq* su *sun.wmq.example.com* elenca le directory di dati e log.
 - b. Controllare le autorizzazioni di *wmquser1* creando un file nella sottodirectory dei dati, aggiungendo del contenuto, leggendolo ed eliminandolo.
3. Accedere a *mars* come *wmq\wmquser2* e ripetere i controlli.
4. Eseguire l'attività successiva, per creare un gestore code per utilizzare i dati condivisi e le directory di log; consultare [“Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo” a pagina 541.](#)

Attività correlate

 [Creazione di un dominio Active Directory e DNS su Windows](#)

 [Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows](#)

 [Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo](#)

 [Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo](#)

Questa sezione illustra come utilizzare l'indicatore `-a` nel comando **crtmqm**. L'indicatore `-a` fornisce al gestore code l'accesso ai relativi file di log e dati su una condivisione file remota utilizzando il gruppo di protezione alternativo.

In una configurazione della scala di produzione, potrebbe essere necessario adattare la configurazione a un dominio esistente. Ad esempio, è possibile definire diversi gruppi di domini per autorizzare diverse condivisioni e per raggruppare gli ID utente che eseguono gestori code.

La configurazione di esempio è composta da tre server:

sun

Un controller di dominio Windows Server 2008. Possiede il dominio *wmq.example.com* che contiene *Sun, mars, venus*. A scopo illustrativo, viene utilizzato anche come server di file.

mars

Un Windows Server 2008 utilizzato come primo server IBM MQ. Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

venus

Un Windows Server 2008 utilizzato come secondo server IBM MQ. Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

Effettuare le operazioni riportate di seguito. Le attività creano il controller di dominio e il dominio, installano IBM MQ for Windows su un server e creano la condivisione file per i file di dati e di log. Se si sta configurando un controller di dominio esistente, potrebbe essere utile provare la procedura su un nuovo server Windows 2008. È possibile adattare la procedura al dominio.

1. [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532.
2. [“Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows”](#) a pagina 536.
3. [“Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows”](#) a pagina 539.

Informazioni su questa attività

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze”](#) a pagina 528.

In questa attività, si crea un gestore code che memorizza i dati e i log in una directory remota su un file server. Ai fini di questo esempio, il server di file è lo stesso server del controller di dominio. La directory contenente le cartelle di dati e di log è condivisa con l'autorizzazione di controllo completo fornita al gruppo globale `wmqha`.

Procedura

1. Accedere al server di dominio, `mars`, come amministratore locale, `mars\Administrator`.
2. Apri una finestra di comando.
3. Riavviare il servizio IBM MQ .

È necessario riavviare il servizio in modo che l'ID utente con cui viene eseguito acquisisca le ulteriori credenziali di sicurezza configurate per esso.

Immettere i comandi:

```
endmqsvc  
strmqsvc
```

Le risposte del sistema:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

E:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. Creare il gestore code.

```
crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs  
QMGR
```

È necessario specificare il dominio, `wmq`, del gruppo di protezione alternativo `wmqha` specificando il nome dominio completo del gruppo globale `"wmq\wmqha"`.

È necessario specificare il nome UNC (Universal Naming Convention) della condivisione `\\sun\wmq` non utilizzare un riferimento unità mappato.

La risposta del sistema:

```
IBM MQ queue manager created.  
Directory '\\sun\wmq\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

Operazioni successive

Verificare il gestore code inserendo e ricevendo un messaggio in una coda.

1. Avviare il gestore code.

```
strmqm QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Creare una coda di test.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

La risposta del sistema:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. Inserire un messaggio di verifica utilizzando il programma di esempio **amqspout**.

```
echo 'A test message' | amqspout QTEST QMGR
```

La risposta del sistema:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. Richiamare il messaggio di test utilizzando il programma di esempio **amqsget**.

```
amqsget QTEST QMGR
```

La risposta del sistema:

```
Sample AMQSGETO start  
message A test message  
Wait 15 seconds ...  
no more messages  
Sample AMQSGETO end
```

5. Chiudere il gestore code.

```
endmqm -i QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended.
```

6. Eliminare il gestore code.

```
dltmqm QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Eliminare le directory create.

Suggerimento: Aggiungere l'opzione /Q ai comandi per evitare che il comando richieda di eliminare ogni file o directory.

```
del /F /S C:\wmq\*. *  
rmdir /S C:\wmq
```

Attività correlate

Windows [Creazione di un dominio Active Directory e DNS su Windows](#)

Windows [Installazione di IBM MQ su un server o su una stazione di lavoro in un dominio Windows](#)

Windows [Creazione di una directory condivisa per i file di log e i dati del gestore code su Windows](#)

Windows [Creazione di un gestore code a più istanze su controller di dominio Windows](#)

Un esempio mostra come configurare un gestore code a più istanze su Windows sui controller di dominio. La configurazione dimostra i concetti coinvolti, piuttosto che essere una scala di produzione. L'esempio è basato su Windows Server 2008. La procedura potrebbe differire su altre versioni di Windows Server.

La configurazione utilizza il concetto di mini - dominio o "domainlet" ; vedere [Windows 2000, Windows Server 2003 e Windows Server 2008 nodi cluster come controller di dominio](#). Per aggiungere gestori code a più istanze a un dominio esistente, consultare ["Creazione di un gestore code a più istanze su workstation o server di dominio su Windows"](#) a pagina 529.

La configurazione di esempio è composta da tre server:

sun

Un server Windows Server 2008 utilizzato come primo controller di dominio. Definisce il dominio *wmq.example.com* che contiene *sun*, *earth* e *mars*. Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

earth

Un server Windows Server 2008 utilizzato come secondo controller di dominio IBM MQ. Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

mars

Un Windows Server 2008 utilizzato come server di file.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

1. Su Windows, non è necessario verificare il file system su cui si intende memorizzare i dati del gestore code e i file di log. La procedura di controllo, [Verifica del funzionamento del file system condiviso](#), è applicabile a AIX and Linux. Su Windows, le verifiche hanno sempre esito positivo.
2. Eseguire le operazioni in [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532 per creare il primo controller di dominio.
3. Eseguire le operazioni in [“Aggiunta di un secondo controller di dominio Windows a un dominio di esempio”](#) a pagina 548 per aggiungere un secondo controller di dominio, installare IBM MQ for Windows su entrambi i controller di dominio e verificare le installazioni.
4. Effettuare le operazioni riportate in [“Installazione di IBM MQ su controller di dominio di Windows in un dominio di esempio”](#) a pagina 550 per installare IBM MQ sui due controller di dominio.

Informazioni su questa attività

Su un file server nello stesso dominio, creare una condivisione per le directory di dati e di log del gestore code. Successivamente, creare la prima istanza di un gestore code a più istanze che utilizzi la condivisione file su uno dei controller di dominio. Creare l'altra istanza sull'altro controller di dominio e infine verificare la configurazione. È possibile creare la condivisione file su un controller di dominio.

Nell'esempio, *sun* è la prima unità di controllo del dominio, *earth* la seconda e *mars* è il server di file.

Procedura

1. Creare le directory che devono contenere i file di log e i dati del gestore code.
 - a) Su *mars*, immettere il comando:

```
md c:\wmq\data , c:\wmq\logs
```

2. Condividere le directory che devono contenere i file di log e i dati del gestore code.

È necessario consentire l'accesso di controllo completo al gruppo locale del dominio *mqme* all'ID utente utilizzato per creare il gestore code. Nell'esempio, gli ID utente membri di *Domain Administrators* hanno l'autorità per creare i gestori code.

La condivisione file deve essere su un server che si trovi nello stesso dominio dei controller di dominio. Nell'esempio, il server *mars* si trova nello stesso dominio dei controller di dominio.

- a) In Windows Explorer, fare clic con il pulsante destro del mouse su **c: \wmq > Proprietà**.
- b) Fare clic sulla scheda **Sicurezza**, quindi su **Avanzate > Modifica ...**
- c) Deselezionare la check box per **Includi autorizzazioni ereditabili dal proprietario di questo oggetto**. Fare clic su **Copia** nella finestra Sicurezza Windows.
- d) Selezionare le linee per gli utenti nell'elenco **Voci di autorizzazione** e fare clic su **Rimuovi**. Lasciare le righe per SYSTEM, Administrators e CREATOR OWNER nell'elenco **Voci di autorizzazione**.

- e) Fare clic su **Aggiungi ...**, e immettere il nome del gruppo locale del dominio *mqm*. Fare clic su **Verifica nomi**
 - f) In risposta a una finestra Sicurezza Windows , immettere il nome e la password di Domain Administrator e fare clic su **OK > OK**.
 - g) Nella finestra Immissione per *wmq* , selezionare **Controllo completo** nell'elenco di **Autorizzazioni**.
 - h) Fare clic su **OK > Applica > OK > OK > OK**
 - i) Ripetere i passi e per h per aggiungere Domain Administrators.
 - j) In Windows Explorer, fare clic con il pulsante destro del mouse su **c: \wmq > Condividi ...**
 - k) Fare clic su **Condivisione avanzata ...** e selezionare la casella di spunta **Condividi questa cartella** . Lasciare il nome condivisione come *wmq*.
 - l) Fare clic su **Autorizzazioni > Aggiungi ...**, e immettere il nome del gruppo locale del dominio *mqm* ; Domain Administrators. Fare clic su **Verifica nomi**.
 - m) In risposta a una finestra Sicurezza Windows , immettere il nome e la password di Domain Administrator e fare clic su **OK > OK**.
3. Creare il gestore code *QMGR* sul primo controller di dominio, *sun*.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

La risposta del sistema:

```
IBM MQ queue manager created.
Directory '\\mars\wmq\data\QMGR' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMGR'.
Default objects statistics : 74 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. Avviare il gestore code su *sun*, consentendo un'istanza in standby.

```
strmqm -x QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. Creare una seconda istanza di *QMGR* su *earth*.

- a) Controllare che i parametri Prefisso e InstallationName siano corretti per *earth*.

In *sun*, eseguire il comando **dspmqinf** :

```
dspmqinf QMGR
```

La risposta del sistema:

```
QueueManager:
Name=QMGR
```

```
Directory=QMGR
Prefix=C:\ProgramData\IBM\MQ
DataPath=\\mars\wmq\data\QMGR
InstallationName=Installation1
```

- b) Copiare il formato leggibile dalla macchina della stanza **QueueManager** negli appunti.

Su *sun* , eseguire nuovamente il comando **dspmqrinf** con il parametro `-o command` .

```
dspmqrinf -o command QMGR
```

La risposta del sistema:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

- c) In *earth* eseguire il comando **addmqinf** dagli appunti per creare un'istanza del gestore code su *earth*.

Modificare il comando, se necessario, per adattare le differenze nei parametri Prefisso o `InstallationName` .

```
addmqinf -s QueueManager -v Name= QMGR
-v Directory= QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

IBM MQ configuration information added.

6. Avviare l'istanza in standby del gestore code in *earth*.

```
strmqm -x QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

Risultati

Verificare che il gestore code passi da *sun* a *earth*:

1. Su *sun*, eseguire il seguente comando:

```
endmqm -i -r -s QMGR
```

La risposta del sistema su *sun*:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
```

IBM MQ queue manager 'QMGR' ended, permitting switchover to a standby instance.

2. Su *earth* immettere ripetutamente il comando:

```
dspmq
```

Le risposte del sistema:

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

Operazioni successive

Per verificare un gestore code a più istanze utilizzando programmi di esempio, consultare [“Verifica del gestore code a più istanze su Windows”](#) a pagina 552.

Attività correlate

[“Aggiunta di un secondo controller di dominio Windows a un dominio di esempio”](#) a pagina 548

[“Installazione di IBM MQ su controller di dominio di Windows in un dominio di esempio”](#) a pagina 550

Informazioni correlate

[Nodi cluster Windows 2000, Windows Server 2003 e Windows Server 2008 come controller di dominio](#)

Windows

Aggiunta di un secondo controller di dominio Windows a un dominio di esempio

Aggiungere un secondo controller di dominio al dominio *wmq.example.com* per creare un dominio Windows in cui eseguire gestori code a più istanze su controller di dominio e server di file.

La configurazione di esempio è composta da tre server:

sun

Un server Windows Server 2008 utilizzato come primo controller di dominio. Definisce il dominio *wmq.example.com* che contiene *sun*, *earth* e *mars*. Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

earth

Un server Windows Server 2008 utilizzato come secondo controller di dominio IBM MQ. Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

mars

Un Windows Server 2008 utilizzato come server di file.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

1. Effettuare le operazioni in [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532 per creare un controller di dominio, *sun*, per il dominio *wmq.example.com*. Modificare i nomi in corsivo per adattarli alla propria configurazione.
2. Installare Windows Server 2008 su un server nel gruppo di lavoro predefinito, WORKGROUP. Per l'esempio, il server è denominato *earth*.

Informazioni su questa attività

In questa attività si configura un server Windows 2008, denominato *earth*, come secondo controller di dominio nel dominio *wmq.example.com*.

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze”](#) a pagina 528.

Procedura

1. Aggiungere il controller di dominio, *sun.wmq.example.com* a *earth* come server DNS.
 - a) Su *earth*, collegarsi come *earth\Administrator* e fare clic su **Avvia**.
 - b) Fare clic con il pulsante destro del mouse su **Rete > Proprietà > Gestisci connessioni di rete**.
 - c) Fare clic con il tasto destro del mouse sull'adattatore di rete, fare clic su **Proprietà**.

Il sistema risponde con la finestra Proprietà connessione area locale che elenca le voci utilizzate dalla connessione.
 - d) Selezionare **Internet Protocol Versione 4** o **Internet Protocol IBM WebSphere MQ 6** dall'elenco di voci nella finestra Proprietà di connessione alla rete locale. Fare clic su **Proprietà > Avanzate ...** e fare clic sul separatore **DNS**.
 - e) Sotto gli indirizzi server DNS, fare clic su **Aggiungi ...**.
 - f) Immettere l'indirizzo IP del controller di dominio, che è anche il server DNS, e fare clic su **Aggiungi**.
 - g) Fare clic su **Aggiungi questi suffissi DNS > Aggiungi**
 - h) Immettere *wmq.example.com* e fare clic su **Aggiungi**.
 - i) Immettere *wmq.example.com* nel campo **Suffisso DNS per questa connessione**.
 - j) Seleziona **Register this connection's address in DNS** e **Use this connection's suffix in DNS registration**. Fare clic su **OK > OK > Chiudi**
 - k) Aprire una finestra comandi e immettere il comando **ipconfig /all** per esaminare le impostazioni TCP/IP.
2. Accedere al controller di dominio, *sun*, come amministratore locale o Workgroup.

Se il server è già configurato come controller di dominio, è necessario collegarsi come amministratore di dominio.
3. Eseguire la procedura guidata Servizi dominio di Active Directory.
 - a) Fare clic su **Avvia > Esegui ...** Immettere *dcprromo* e fare clic su **OK**.

Se i file binari di Active Directory non sono già installati, Windows installa automaticamente i file.
4. Configurare *earth* come secondo controller di dominio nel dominio *wmq.example.com*.
 - a) Nella prima finestra della procedura guidata, lasciare deselezionata la check box **Utilizza installazione in modalità avanzata**. Fare clic su **Avanti > Avanti** e fare clic su **Crea Aggiungi un controller di dominio a un dominio esistente > Avanti**.
 - b) Immettere *wmq* in **Immettere il nome di qualsiasi dominio in questa struttura ...**. Si fa clic sul pulsante di opzione **Credenziali alternative**, quindi su **Imposta ...** Immettere il nome e la password dell'amministratore del dominio e fare clic su **OK > Avanti > Avanti > Avanti**.
 - c) Nella finestra Opzioni aggiuntive controller di dominio, accettare le opzioni **Server DNS** e **Catalogo globale**, che sono selezionate; fare clic su **Avanti > Avanti**.
 - d) In Directory Services Restore Mode Administrator Password, immettere **Password** e **Conferma password** e fare clic su **Avanti > Avanti**.
 - e) Quando viene richiesto **Credenziali di rete**, immettere la password dell'amministratore del dominio. Selezionare **Riavvia al completamento** nella finestra finale della procedura guidata.
 - f) Dopo un po' di tempo, potrebbe aprirsi una finestra con un errore **DCPrromo** relativo alla delega DNS; fare clic su **OK**. Il server viene riavviato.

Risultati

Una volta riavviato *earth*, accedere come amministratore del dominio. Verificare che il dominio *wmq.example.com* sia stato replicato in *earth*.

Operazioni successive

Continuare con l'installazione di IBM MQ ; consultare [“Installazione di IBM MQ su controller di dominio di Windows in un dominio di esempio”](#) a pagina 550.

Attività correlate

Windows [Installazione di IBM MQ su controller di dominio di Windows in un dominio di esempio “Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532

Windows [Installazione di IBM MQ su controller di dominio di Windows in un dominio di esempio](#)
Installare e configurare le installazioni di IBM MQ su entrambi i controller di dominio nel dominio *wmq.example.com*.

La configurazione di esempio è composta da tre server:

sun

Un server Windows Server 2008 utilizzato come primo controller di dominio. Definisce il dominio *wmq.example.com* che contiene *sun*, *earth*, *mars*. Contiene un'istanza del gestore code a più istanze denominato *QMGR*.

earth

Un server Windows Server 2008 utilizzato come secondo controller di dominio IBM MQ . Contiene la seconda istanza del gestore code a più istanze denominato *QMGR*.

mars

Un Windows Server 2008 utilizzato come server di file.

Sostituire i nomi in corsivo nell'esempio, con nomi di propria scelta.

Prima di iniziare

1. Effettuare le operazioni in [“Creazione di un dominio Active Directory e DNS su Windows”](#) a pagina 532 per creare un controller di dominio, *sun*, per il dominio *wmq.example.com*. Modificare i nomi in corsivo per adattarli alla propria configurazione.
2. Effettuare le operazioni riportate in [“Aggiunta di un secondo controller di dominio Windows a un dominio di esempio”](#) a pagina 548 per creare un secondo controller di dominio, *earth*, per il dominio *wmq.example.com*. Modificare i nomi in corsivo per adattarli alla propria configurazione.
3. Consultare [Requisiti hardware e software sui sistemi Windows](#) per altre versioni Windows su cui è possibile eseguire IBM MQ .

Informazioni su questa attività

Installare e configurare le installazioni di IBM MQ su entrambi i controller di dominio nel dominio *wmq.example.com*.

Procedura

1. Installare IBM MQ su *sun* e *earth*.

Per ulteriori informazioni, consultare [Installazione del server IBM MQ su Windows](#) .

- a) Su *sun* e *earth*, accedere come amministratore del dominio, *wmq\Administrator*.
- b) Eseguire il comando **Setup** sul supporto di installazione IBM MQ for Windows .
Viene avviata l'applicazione IBM MQ Launchpad.
- c) Fare clic su **Requisiti software** per verificare che il software prerequisito sia installato.
- d) Fare clic su **Configurazione di rete > No**.

È possibile configurare o meno un ID utente di dominio per questa installazione. L'ID utente creato è un ID utente locale del dominio.

- e) Fare clic su **IBM MQ Installazione**, selezionare una lingua di installazione e fare clic su Avvia IBM MQ Installer.
- f) Confermare l'accordo di licenza e fare clic su **Avanti > Avanti > Installa** per accettare la configurazione predefinita. Attendere il completamento dell'installazione e fare clic su **Fine**.

Se si desidera modificare il nome dell'installazione, installare componenti differenti, configurare una directory differente per i dati e i log del gestore code oppure eseguire l'installazione in una directory diversa, fare clic su **Personalizzato** anziché su **Tipico**.

IBM MQ è installato e il programma di installazione avvia Prepare IBM MQ Wizard.

L'installazione di IBM MQ for Windows configura un gruppo locale di dominio mqme un gruppo di dominio Domain mqm. Rende Domain mqm un membro di mqm. I controller di dominio successivi nello stesso dominio condividono i gruppo mqm e Domain mqm .

2. Su *earth* e *sun*, eseguire Prepare IBM MQ Wizard.

Per ulteriori informazioni, consultare [Configurazione di IBM MQ con Prepare IBM MQ Wizard](#).

- a) Il programma di installazione di IBM MQ esegue automaticamente Prepare IBM MQ Wizard .

Per avviare manualmente la procedura guidata, individuare il collegamento alla cartella Prepare IBM MQ Wizard nella cartella **Start > Tutti i programmi > IBM MQ** . Selezionare il collegamento che corrisponde all'installazione di IBM MQ in una configurazione a più installazioni.

- b) Fare clic su **Avanti** e lasciare **No** selezionato in risposta alla domanda "Identifica se è presente un controller di dominio Windows 2000 o successivo nella rete"¹.
- c) Nella pagina finale della procedura guidata, selezionare o deselezionare le caselle di spunta richieste e fare clic su **Fine**.

Il Prepare IBM MQ Wizard crea un utente locale di dominio MUSR_MQADMIN sul primo controller di dominio e un altro utente locale di dominio MUSR_MQADMIN1 sul secondo controller di dominio. La procedura guidata crea il servizio IBM MQ su ciascun controllore, con MUSR_MQADMIN o MUSR_MQADMIN1 come utente che accede al servizio.

3. Definire un utente che dispone dell'autorizzazione per creare un gestore code.

L'utente deve avere il diritto di accedere localmente e deve essere membro del gruppo mqm locale del dominio. Sui controller di dominio, gli utenti di dominio non hanno il diritto di accedere localmente, ma gli amministratori sì. Per impostazione predefinita, nessun utente ha entrambi questi attributi. In questa attività, aggiungere amministratori di dominio al gruppo mqm locale del dominio.

- a) Aprire **Server Manager > Ruoli > Active Directory Domain Services > wmq.example.com > Utenti**.
- b) Fare clic con il tasto destro del mouse su **Admin dominio > Aggiungi a un gruppo ...** e immettere mqm ; fare clic su **Verifica nomi > OK > OK**

Risultati


1. Verificare che Prepare IBM MQ Wizard abbia creato l'utente del dominio, MUSR_MQADMIN:
 - a. Aprire **Server Manager > Ruoli > Active Directory Domain Services > wmq.example.com > Utenti**.
 - b. Fare clic con il tasto destro del mouse su **MUSR_MQADMIN > Proprietà ... > Membro** die vedere che è un membro di Domain users e mqm.
2. Verificare che MUSR_MQADMIN abbia il diritto di essere eseguito come servizio:
 - a. Fare clic su Fare clic su **Avvia > Esegui ...**, immettere il comando **secpol.msc** e fare clic su **OK**.
 - b. Aprire **Impostazioni di protezione > Politiche locali > Assegnazione diritti utente**. Nell'elenco delle politiche, fare clic con il tasto destro del mouse su **Accedi come servizio > Proprietà** e vedi MUSR_MQADMIN è elencato come avente il diritto di accedere come un servizio. Fare clic su **OK**.

¹ È possibile configurare l'installazione per il dominio. Poiché tutti gli utenti e i gruppi su un controller di dominio hanno un ambito di dominio, non fa alcuna differenza. È più semplice installare IBM MQ come se non fosse nel dominio.

Operazioni successive


1. Eseguire l'attività, [“Lettura e scrittura di dati e file di log autorizzati dal gruppo mqm locale”](#) a pagina 559, per verificare che l'installazione e la configurazione stiano funzionando correttamente.
2. Tornare all'attività, [“Creazione di un gestore code a più istanze su controller di dominio Windows”](#) a pagina 544, per completare l'attività di configurazione di un gestore code a più istanze sui controller di dominio.

Attività correlate

 [Aggiunta di un secondo controller di dominio Windows a un dominio di esempio](#)

Riferimenti correlati

[Diritti utente richiesti per un servizio IBM MQ Windows](#)

 [Verifica del gestore code a più istanze su Windows](#)

Utilizzare i programmi di esempio **amqsgbac**, **amqsphac** e **amqsmhac** per verificare la configurazione di un gestore code a più istanze. Questo argomento fornisce una configurazione di esempio per verificare una configurazione del gestore code a più istanze su Windows Server 2003.

I programmi di esempio ad alta disponibilità utilizzano la riconnessione client automatica. Quando il gestore code connesso ha esito negativo, il client tenta di riconnettersi a un gestore code nello stesso gruppo di gestori code. La descrizione degli esempi, [Programmi di esempio ad alta disponibilità](#), illustra la riconnessione del client utilizzando un gestore code a istanza singola per semplicità. È possibile utilizzare gli stessi esempi con i gestori code a più istanze per verificare una configurazione del gestore code a più istanze.

Questo esempio utilizza la configurazione a più istanze descritta in [“Creazione di un gestore code a più istanze su controller di dominio Windows”](#) a pagina 544. Utilizzare la configurazione per verificare che il gestore code a più istanze passi all'istanza in standby. Arrestare il gestore code con il comando **endmqm** e utilizzare l'opzione **-s, switchover**. I programmi client si riconnettono alla nuova istanza del gestore code e continuano a lavorare con la nuova istanza dopo un leggero ritardo.

Il client è installato in un'immagine VMware da 400 MB su cui è in esecuzione Windows 7 Service Pack 1. Per motivi di sicurezza, è connesso sulla stessa rete solo host VMware dei server di dominio che eseguono il gestore code a più istanze. Condivide la cartella **/MQHA**, che contiene la tabella di connessione client, per semplificare la configurazione.

Verifica del failover utilizzando IBM MQ Explorer

Prima di utilizzare le applicazioni di esempio per verificare il failover, eseguire IBM MQ Explorer su ciascun server. Aggiungere entrambe le istanze del gestore code a ciascun explorer utilizzando la procedura guidata **Aggiungi gestore code remoto > Connetti direttamente a gestore code a più istanze**. Verificare che entrambe le istanze siano in esecuzione, consentendo lo standby. Chiudere la finestra eseguendo l'immagine VMware con l'istanza attiva, spegnendo virtualmente il server o arrestando l'istanza attiva, consentendo la commutazione dell'istanza in standby e la riconnessione dei client.



Attenzione: Se si spegne il server, assicurarsi che non sia quello che ospita la cartella MQHA !

Nota: L'opzione **Consenti commutazione a un'istanza in standby** potrebbe non essere disponibile nella finestra di dialogo **Arresta gestore code**. L'opzione è mancante perché il gestore code è in esecuzione come gestore code a istanza singola. È necessario che sia stato avviato senza l'opzione **Consenti un'istanza standby**. Se la tua richiesta di arrestare il gestore code viene rifiutata, guarda la finestra **Details**, probabilmente non c'è alcuna istanza in standby in esecuzione.

Verifica del failover utilizzando i programmi di esempio

Scegliere un server per eseguire l'istanza attiva

È possibile che sia stato scelto uno dei server per ospitare la directory o il file system MQHA. Se si prevede di verificare il failover chiudendo la finestra VMware che esegue il server attivo, assicurarsi che non sia quello che ospita MQHA !

Sul server che esegue l'istanza del gestore code attivo

1. Modificare *ipaddr1* e *ipaddr2* e salvare i seguenti comandi in `N:\hasample.tst`.

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(' ipaddr1 (1414), ipaddr2 (1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

Nota: Lasciando vuoto il parametro **MCAUSER**, l'ID utente client viene inviato al server. L'ID utente client deve avere le autorizzazioni corrette sui server. Un'alternativa è quella di impostare il parametro **MCAUSER** nel canale SVRCONN sull'ID utente configurato sul server.

2. Aprire un prompt dei comandi con il percorso `N:\` ed eseguire il comando:

```
runmqsc -m QM1 < hasample.tst
```

3. Verificare che il listener sia in esecuzione e che disponga del controllo del gestore code, ispezionando l'output del comando **runmqsc**.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

Oppure, utilizzando il IBM MQ Explorer che il listener TCP/IP è in esecuzione e dispone di `Control = Queue Manager`.

Sul client

1. Associare la directory condivisa `C:\MQHA` sul server a `N:\` sul client.
2. Aprire un prompt dei comandi con il percorso `N:\`. Impostare la variabile di ambiente `MQCHLLIB` in modo che punti alla tabella di definizione del canale client (CCDT) sul server:

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. Al prompt dei comandi immettere i seguenti comandi:

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

Nota: In caso di problemi, avviare le applicazioni da un prompt dei comandi in modo che il codice di errore venga stampato sulla console oppure consultare `AMQERR01.LOG` nella cartella `N:\data\QM1\errors`.

Sul server che esegue l'istanza del gestore code attivo

1. Le alternative sono:
 - Chiudere la finestra che esegue l'immagine VMware con l'istanza del server attiva.
 - Utilizzando IBM MQ Explorer, arrestare l'istanza del gestore code attivo, consentendo il passaggio all'istanza in standby e istruendo i client riconnettibili a riconnettersi.
2. I tre client alla fine rilevano che la connessione è interrotta e quindi si ricollegano. In questa configurazione, se si chiude la finestra del server, sono necessari circa sette minuti per ristabilire tutte e tre le connessioni. Alcune connessioni vengono ristabilite ben prima di altre.

Risultati

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Windows Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows

Questo argomento descrive come proteggere un'ubicazione condivisa per i dati del gestore code e i file di log utilizzando un gruppo di sicurezza alternativo globale. È possibile condividere l'ubicazione tra diverse istanze di un gestore code in esecuzione su server differenti.

Generalmente, non si imposta un'ubicazione condivisa per i dati del gestore code e i file di log. Quando si installa IBM MQ for Windows, il programma di installazione crea una directory home di propria scelta per tutti i gestori code creati su tale server. Protegge le directory con il gruppo mqm locale e configura un ID utente per il servizio di IBM MQ per accedere alle directory.

Quando si protegge una cartella condivisa con un gruppo di sicurezza, un utente a cui è consentito accedere alla cartella deve disporre delle credenziali del gruppo. Si supponga che una cartella su un server di file remoto sia protetta con il gruppo mqm locale su un server denominato *mars*. Rendere l'utente che esegue i processi del gestore code un membro del gruppo mqm locale su *mars*. L'utente dispone di credenziali che corrispondono a quelle della cartella sul file server remoto. Utilizzando queste credenziali, il gestore code è in grado di accedere ai propri dati e file di log nella cartella. L'utente che esegue i processi del gestore code su un server differente è membro di un gruppo mqm locale diverso che non dispone di credenziali corrispondenti. Quando il gestore code viene eseguito su un altro server su *mars*, non può accedere ai dati e ai file di log che ha creato quando è stato eseguito su *mars*. Anche se si rende l'utente un utente di dominio, ha credenziali diverse, perché deve acquisire le credenziali dal gruppo mqm locale su *mars* e non può farlo da un server diverso.

Fornire al gestore code un gruppo di sicurezza alternativo globale risolve il problema; consultare [Figura 73 a pagina 555](#). Proteggere una cartella remota con un gruppo globale. Passare il nome del gruppo globale al gestore code quando lo si crea su *mars*. Passare il nome del gruppo globale come gruppo di protezione alternativo utilizzando il parametro `-a [r]` sul comando **crtmqm**. Se si trasferisce il gestore code per l'esecuzione su un server differente, il nome del gruppo di protezione viene trasferito con esso. Il nome viene trasferito nella stanza **AccessMode** nel file `qm.ini` come `SecurityGroup`; ad esempio:

```
AccessMode:
SecurityGroup=wmq\wmq
```

La sezione **AccessMode** in `qm.ini` include anche `RemoveMQMAccess`; ad esempio:

```
AccessMode:
RemoveMQMAccess=true/false
```

Se questo attributo viene specificato con il valore `true` ed è stato fornito anche un gruppo di accesso, al gruppo `mqm` locale non viene concesso l'accesso ai file di dati del gestore code.

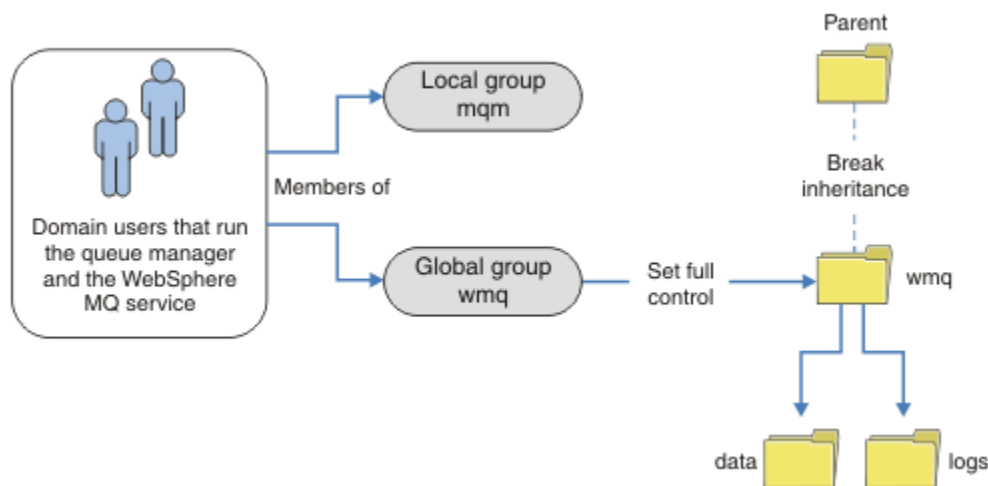


Figura 73. Protezione dei dati e dei log del gestore code mediante un gruppo di sicurezza globale alternativo (1)

Per l'ID utente con cui devono essere eseguiti i processi del gestore code per avere le credenziali corrispondenti del gruppo di sicurezza globale, l'ID utente deve avere anche un ambito globale. Non è possibile rendere un gruppo locale o un principal un membro di un gruppo globale. In [Figura 73 a pagina 555](#), gli utenti che eseguono i processi del gestore code vengono visualizzati come utenti del dominio.

Se si stanno distribuendo molti server IBM MQ, il raggruppamento di utenti in [Figura 73 a pagina 555](#) non è conveniente. È necessario ripetere il processo di aggiunta di utenti ai gruppi locali per ogni server IBM MQ. Invece, creare un gruppo globale `Domain mqm` sul controller di dominio e rendere gli utenti che eseguono IBM MQ membri del gruppo `Domain mqm`; consultare [Figura 74 a pagina 556](#). Quando si installa IBM MQ come installazione di dominio, il Prepare IBM MQ Wizard rende automaticamente il gruppo `Domain mqm` membro del gruppo `mqm` locale. Gli stessi utenti si trovano in entrambi i gruppi globali `Domain mqm` e `wmq`.

Suggerimento: Gli stessi utenti possono eseguire IBM MQ su server differenti, ma su un singolo server è necessario disporre di utenti differenti per eseguire IBM MQ come servizio ed eseguire in modo interattivo. È inoltre necessario disporre di utenti differenti per ogni installazione su un server. Di solito, quindi, `Domain mqm` contiene un numero di utenti.

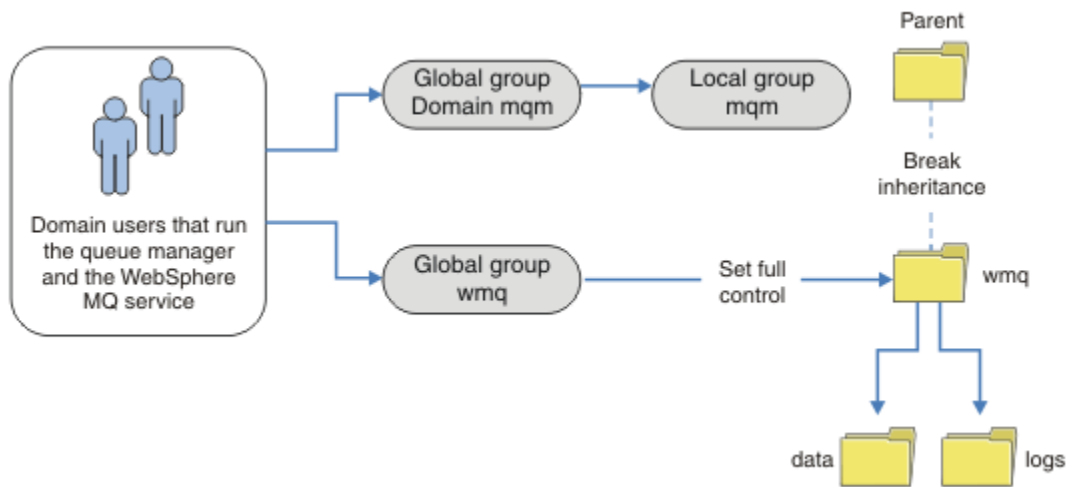


Figura 74. Protezione dei dati e dei log del gestore code utilizzando un gruppo di sicurezza globale alternativo (2)

L'organizzazione in Figura 74 a pagina 556 è inutilmente complicata così com'è. L'accordo ha due gruppi globali con membri identici. È possibile semplificare l'organizzazione e definire un solo gruppo globale; consultare Figura 75 a pagina 556.

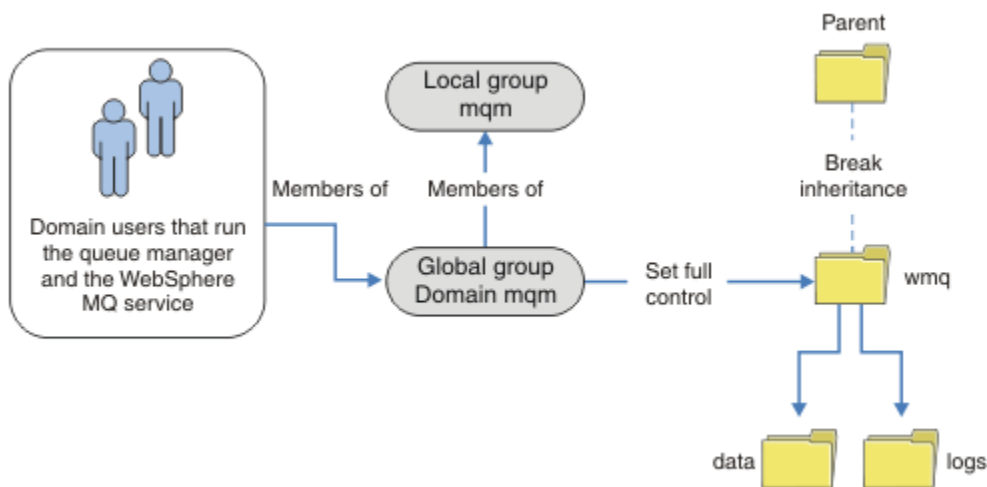


Figura 75. Protezione dei log e dei dati del gestore code utilizzando un gruppo di sicurezza globale alternativo (3)

In alternativa, potrebbe essere necessario un livello più elevato di controllo degli accessi, con diversi gestori code limitati ad accedere a cartelle differenti; consultare Figura 76 a pagina 557. In Figura 76 a pagina 557, sono definiti due gruppi di utenti del dominio, in gruppi globali separati per proteggere file di dati e di log del gestore code differenti. Vengono visualizzati due diversi gruppi mqm locali, che devono trovarsi su server IBM MQ diversi. In questo esempio, i gestori code sono suddivisi in due insiemi, con utenti differenti assegnati ai due insiemi. I due set potrebbero essere gestori code di test e di produzione. I gruppi di sicurezza alternativi sono denominati wmq1 e wmq2. È necessario aggiungere manualmente i gruppi globali wmq1 e wmq2 ai gestori code corretti in base al fatto che si trovino nel dipartimento di test o di produzione. La configurazione non può trarre vantaggio dal fatto che l'installazione di IBM MQ propaga Domain mqm al gruppo mqm locale come in Figura 75 a pagina 556, poiché esistono due gruppi di utenti.

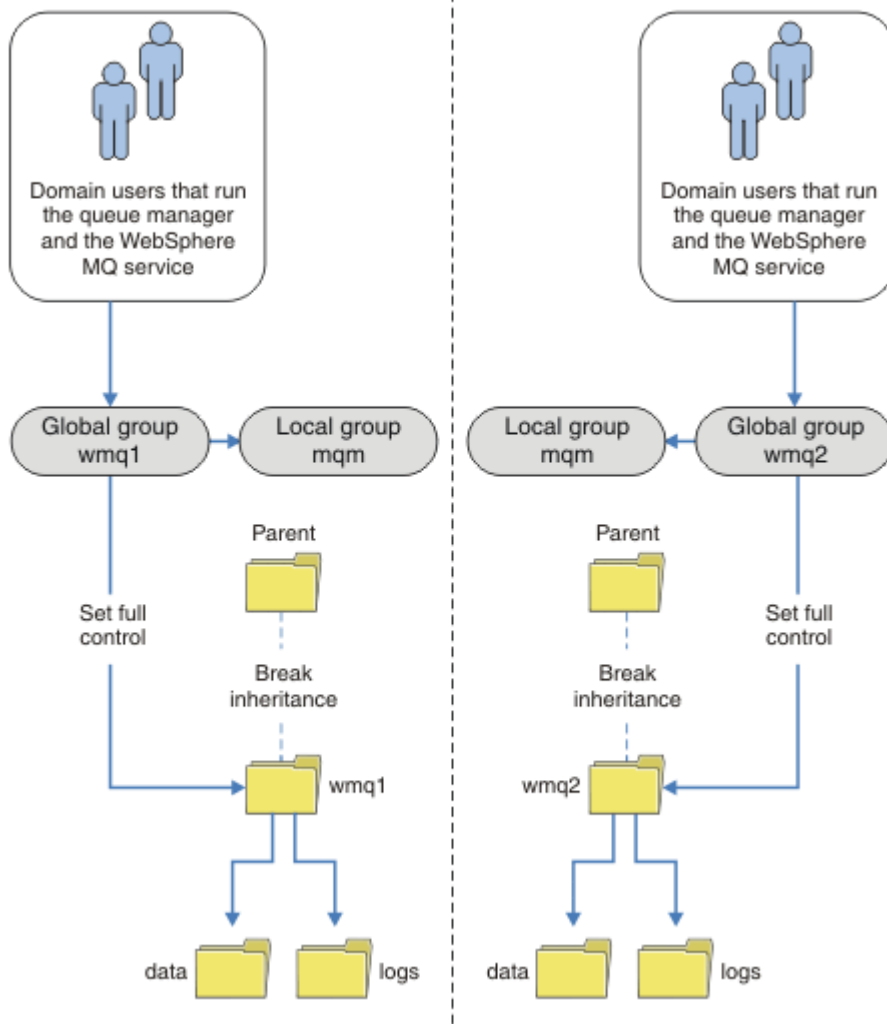


Figura 76. Protezione dei dati e dei log del gestore code utilizzando un principal di sicurezza globale alternativo (4)

Un modo alternativo per partizionare due dipartimenti sarebbe quello di posizionarli in due domini Windows. In tal caso, è possibile tornare a utilizzare il modello più semplice mostrato in [Figura 75](#) a pagina 556.

Windows *Proteggere i file e le directory di log e i dati del gestore code non condivisi su Windows*

Questo argomento descrive come proteggere un'ubicazione alternativa per i dati del gestore code e i file di log, sia utilizzando il gruppo mqm locale che un gruppo di protezione alternativo.

In genere non si imposta un'ubicazione alternativa per i dati del gestore code e i file di log. Quando si installa IBM MQ for Windows, il programma di installazione crea una directory home a scelta per tutti i gestori code creati. Protegge le directory con il gruppo mqm locale e configura un ID utente per il servizio di IBM MQ per accedere alle directory.

Due esempi dimostrano come configurare il controllo accessi per IBM MQ. Gli esempi mostrano come creare un gestore code con i relativi dati e log nelle directory che non si trovano nei percorsi di dati e log creati dall'installazione. Nel primo esempio, [“Lettura e scrittura di dati e file di log autorizzati dal gruppo mqm locale”](#) a pagina 559, si consente l'accesso alle directory di coda e log autorizzando il gruppo mqm locale. Il secondo esempio, [“Lettura e scrittura dei dati e dei file di log autorizzati da un gruppo di sicurezza locale alternativo”](#) a pagina 562, differisce in quanto l'accesso alle indirizzari è autorizzato da un gruppo di protezione alternativo. Quando si accede alle directory da un gestore code in esecuzione su un solo server, la protezione dei dati e dei file di log con il gruppo di sicurezza alternativo consente di

proteggere diversi gestori code con diversi gruppi o principal locali. Quando si accede alle directory da un gestore code in esecuzione su server diversi, ad esempio con un gestore code a più istanze, la protezione dei dati e dei file di log con il gruppo di protezione alternativo è l'unica scelta; consultare [“Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows”](#) a pagina 554.

La configurazione delle permessi di sicurezza dei dati del gestore code e dei file di log non è un'attività comune su Windows. Quando si installa IBM MQ for Windows, è possibile specificare le directory per i dati e i log del gestore code oppure accettare le directory predefinite. Il programma di installazione protegge automaticamente queste directory con il gruppo mqm locale, fornendo l'autorizzazione di controllo completo. Il processo di installazione verifica che l'ID utente che esegue i gestori code sia un membro del gruppo mqm locale. È possibile modificare le altre autorizzazioni di accesso sulle directory per soddisfare le proprie esigenze di accesso.

Se si sposta la directory dei file di dati e di log in nuove ubicazioni, è necessario configurare la sicurezza delle nuove ubicazioni. È possibile modificare l'ubicazione delle directory se si esegue il back up di un gestore code e lo si ripristina su un altro computer o se si modifica il gestore code in un gestore code a più istanze. È possibile scegliere tra due modi per proteggere i dati del gestore code e le directory di log nella nuova posizione. È possibile proteggere le directory limitando l'accesso al gruppo mqm locale oppure è possibile limitare l'accesso a qualsiasi gruppo di sicurezza di propria scelta.

È necessario il minor numero di passi per proteggere le directory utilizzando il gruppo mqm locale. Impostare le autorizzazioni sulle directory di dati e log per consentire il controllo completo del gruppo mqm locale. Un approccio tipico consiste nel copiare la serie di autorizzazioni esistenti, rimuovendo l'ereditarietà dal parent. È quindi possibile rimuovere o limitare le autorizzazioni di altri principal.

Se si esegue il gestore code con un ID utente diverso per il servizio impostato dalla procedura guidata Prepara IBM MQ, tale ID utente deve essere membro del gruppo mqm locale. L'attività, [“Lettura e scrittura di dati e file di log autorizzati dal gruppo mqm locale”](#) a pagina 559, consente di eseguire le fasi.

È anche possibile proteggere i dati e i file di log del gestore code utilizzando un gruppo di protezione alternativo. Il processo di protezione dei dati del gestore code e dei file di log con il gruppo di sicurezza alternativo prevede una serie di fasi che fanno riferimento a [Figura 77 a pagina 558](#). Il gruppo locale, wmq, è un esempio di un gruppo di protezione alternativo.

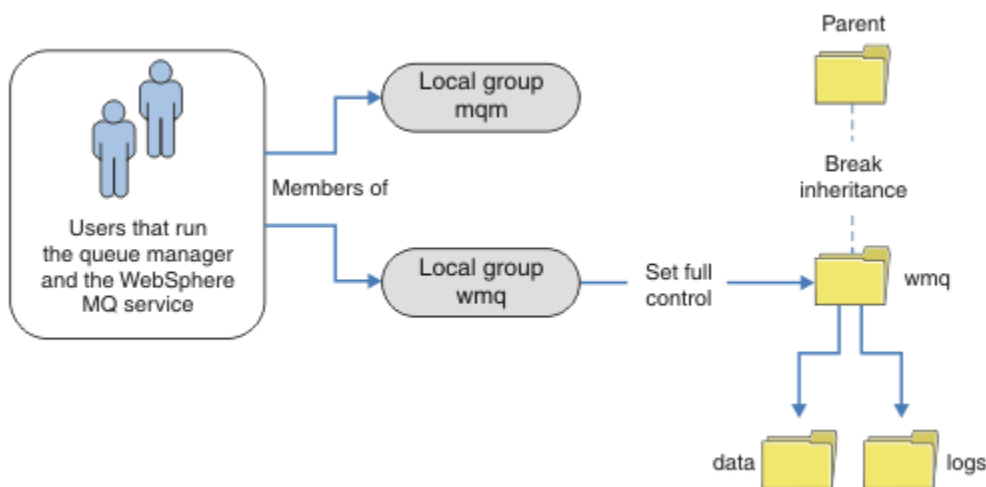


Figura 77. Protezione dei dati e dei log del gestore code utilizzando un gruppo di protezione locale alternativo, wmq

1. Creare directory separate per i dati e i log del gestore code, una directory comune o una directory principale comune.
2. Copiare la serie esistente di autorizzazioni ereditate per le directory o la directory principale e modificarle in base alle proprie esigenze.

3. Proteggere le directory che devono contenere il gestore code e i log fornendo al gruppo alternativo, `wmq`, l'autorizzazione di controllo completo per le directory.
4. Fornire a tutti gli ID utente che eseguono i processi del gestore code le credenziali del principal o del gruppo di sicurezza alternativo:
 - a. Se si definisce un utente come principal di sicurezza alternativo, l'utente deve essere lo stesso utente con cui verrà eseguito il gestore code. L'utente deve essere un membro del gruppo `mqm` locale.
 - b. Se si definisce un gruppo locale come gruppo di sicurezza alternativo, aggiungere l'utente sotto il quale verrà eseguito il gestore code al gruppo alternativo. L'utente deve anche essere membro del gruppo `mqm` locale.
 - c. Se si definisce un gruppo globale come gruppo di sicurezza alternativo, consultare [“Protezione dei dati del gestore code condiviso e delle directory di log e dei file su Windows”](#) a pagina 554.
5. Creare il gestore code specificando il principal o il gruppo di sicurezza alternativo nel comando `crtmqm`, con il parametro `-a`.

Windows *Lettura e scrittura di dati e file di log autorizzati dal gruppo `mqm` locale*

L'attività illustra come creare un gestore code con i relativi dati e file di log memorizzati in qualsiasi directory di propria scelta. L'accesso ai file è protetto dal gruppo `mqm` locale. La directory non è condivisa.

Prima di iniziare

1. Installare IBM MQ for Windows come installazione primaria.
2. Eseguire Prepare IBM MQ Wizard.

Per ulteriori informazioni, consultare [Configurazione di IBM MQ con Prepare IBM MQ Wizard](#).

Per questa attività, configurare l'installazione in modo che venga eseguita con un ID utente locale o con un ID utente di dominio. Alla fine, per completare tutte le attività in [“Domini Windows e gestori code a più istanze”](#) a pagina 528, l'installazione deve essere configurata per un dominio.

3. Accedere con autorità di amministratore per eseguire la prima parte dell'attività.

Informazioni su questa attività

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze”](#) a pagina 528.

In Windows, è possibile creare i percorsi di dati e log predefiniti per un IBM MQ for Windows in qualsiasi directory di propria scelta. La procedura guidata di installazione e configurazione fornisce automaticamente al gruppo `mqm` locale, e all'ID utente che sta eseguendo i processi del gestore code, l'accesso alle directory. Se si crea un gestore code specificando directory differenti per i dati del gestore code e i file di log, è necessario configurare l'autorizzazione di controllo completo per le directory.

In questo esempio, si fornisce al gestore code il controllo completo sui relativi file di dati e di log fornendo al gruppo `mqm` locale l'autorizzazione alla directory `c : \wmq`.

Il comando `crtmqm` crea un gestore code che viene avviato automaticamente quando la workstation viene avviata utilizzando il servizio IBM MQ.

L'attività è illustrativa; utilizza valori specifici che è possibile modificare. I valori che è possibile modificare sono in corsivo. Alla fine dell'attività ..., seguire le istruzioni per rimuovere tutte le modifiche apportate.

Procedura

1. Aprire un prompt dei comandi.
2. Immettere il comando:

```
md c:\wmq\data, c:\wmq\logs
```

3. Impostare le autorizzazioni sulle directory per consentire l'accesso in lettura e scrittura del gruppo mqm locale.

```
cacls c:\wmq/T /E /G mqm:F
```

La risposta del sistema:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. Opzionale: Passare a un ID utente che è un membro del gruppo mqm locale.

È possibile continuare come amministratore, ma per una configurazione di produzione realistica, continuare con un ID utente con diritti più limitati. L'ID utente deve essere almeno un membro del gruppo mqm locale.

Se l'installazione di IBM MQ è configurata come parte di un dominio, rendere l'ID utente un membro del gruppo Domain mqm. La procedura guidata "Prepara IBM MQ" rende il gruppo globale Domain mqm un membro del gruppo mqm locale, quindi non è necessario rendere l'ID utente direttamente un membro del gruppo mqm locale.

5. Creare il gestore code.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

La risposta del sistema:

```
IBM MQ queue manager created.
Directory 'c:\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

6. Verificare che le directory create dal gestore code si trovino nella directory c:\wmq.

```
dir c:\wmq/D /B /S
```

7. Verificare che i file dispongano dell'autorizzazione di lettura e scrittura o di controllo completo per il gruppo mqm locale.

```
cacls c:\wmq\*.*
```

Operazioni successive

Verificare il gestore code inserendo e ricevendo un messaggio in una coda.

1. Avviare il gestore code.

```
strmqm QMGR
```

La risposta del sistema:


```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Creare una coda di test.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

La risposta del sistema:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. Inserire un messaggio di verifica utilizzando il programma di esempio **amqsput**.

```
echo 'A test message' | amqsput QTEST QMGR
```

La risposta del sistema:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. Richiamare il messaggio di test utilizzando il programma di esempio **amqsget**.

```
amqsget QTEST QMGR
```

La risposta del sistema:

```
Sample AMQSGET0 start  
message A test message  
Wait 15 seconds ...  
no more messages  
Sample AMQSGET0 end
```

5. Chiudere il gestore code.

```
endmqm -i QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended.
```

6. Eliminare il gestore code.

```
dltmqm QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Eliminare le directory create.

Suggerimento: Aggiungere l'opzione /Q ai comandi per evitare che il comando richieda di eliminare ogni file o directory.


```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

Concetti correlati

[“Domini Windows e gestori code a più istanze” a pagina 528](#)

Un gestore code a più istanze su Windows richiede che i relativi dati e log siano condivisi. La condivisione deve essere accessibile a tutte le istanze del gestore code in esecuzione su server o workstation differenti. Configurare i gestori code e condividere come parte di un dominio Windows . Il gestore code può essere eseguito su una stazione di lavoro o su un server di dominio o sul controller di dominio.


Attività correlate

 [Lettura e scrittura dei dati e dei file di log autorizzati da un gruppo di sicurezza locale alternativo](#)

Questa sezione illustra come utilizzare l'indicatore -a nel comando **crtmqm** . L'indicatore fornisce al gestore code un gruppo di protezione locale alternativo per fornire l'accesso ai file di log e di dati.

[“Lettura e scrittura di dati condivisi e file di log autorizzati da un gruppo di sicurezza globale alternativo” a pagina 541](#)

[“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows” a pagina 529](#)

 [Lettura e scrittura dei dati e dei file di log autorizzati da un gruppo di sicurezza locale alternativo](#)

Questa sezione illustra come utilizzare l'indicatore -a nel comando **crtmqm** . L'indicatore fornisce al gestore code un gruppo di protezione locale alternativo per fornire l'accesso ai file di log e di dati.

Prima di iniziare

1. Installare IBM MQ for Windows come installazione primaria.
2. Eseguire Prepare IBM MQ Wizard.

Per ulteriori informazioni, consultare [Configurazione di IBM MQ con Prepare IBM MQ Wizard](#).

Per questa attività, configurare l'installazione in modo che venga eseguita con un ID utente locale o con un ID utente di dominio. Alla fine, per completare tutte le attività in [“Domini Windows e gestori code a più istanze” a pagina 528](#), l'installazione deve essere configurata per un dominio.

3. Accedere con autorità di amministratore per eseguire la prima parte dell'attività.

Informazioni su questa attività

Questa attività è una delle attività correlate che illustrano l'accesso ai dati del gestore code e ai file di log. Le attività mostrano come creare un gestore code autorizzato a leggere e scrivere dati e file di log memorizzati in una directory di propria scelta. Accompagnano l'attività, [“Domini Windows e gestori code a più istanze” a pagina 528](#).

In Windows, è possibile creare i percorsi di dati e log predefiniti per un IBM MQ for Windows in qualsiasi directory di propria scelta. La procedura guidata di installazione e configurazione fornisce

automaticamente al gruppo `mqm` locale, e all'ID utente che sta eseguendo i processi del gestore code, l'accesso alle directory. Se si crea un gestore code specificando directory differenti per i dati del gestore code e i file di log, è necessario configurare l'autorizzazione di controllo completo per le directory.

In questo esempio, si fornisce al gestore code un gruppo locale di protezione alternativo che dispone dell'autorizzazione di controllo completo per le directory. Il gruppo di sicurezza alternativo fornisce al gestore code l'autorizzazione a gestire i file nella directory. Lo scopo principale del gruppo di sicurezza alternativo è di autorizzare un gruppo globale di sicurezza alternativo. Utilizzare un gruppo globale di sicurezza alternativo per configurare un gestore code a più istanze. In questo esempio, si configura un gruppo locale per familiarizzare con l'utilizzo di un gruppo di protezione alternativo senza installare IBM MQ in un dominio. È insolito configurare un gruppo locale come gruppo di sicurezza alternativo.

L'attività è illustrativa; utilizza valori specifici che è possibile modificare. I valori che è possibile modificare sono in corsivo. Alla fine dell'attività ..., seguire le istruzioni per rimuovere tutte le modifiche apportate.

Procedura

1. Configurare un gruppo di sicurezza alternativo.

Il gruppo di sicurezza alternativo è generalmente un gruppo di domini. Nell'esempio, si crea un gestore code che utilizza un gruppo di sicurezza alternativo locale. Con un gruppo di protezione locale alternativo, è possibile eseguire l'attività con un'installazione IBM MQ che non fa parte di un dominio.

- a) Eseguire il comando **lusrmgr.msc** per aprire la finestra Utenti e gruppi locali.
- b) Fare clic con il tasto destro del mouse su **Gruppi > Nuovo gruppo ...**
- c) Nel campo **Nome gruppo**, immettere *al_tmqm* e fare clic su **Crea > Chiudi**.
- d) Identifica l'ID utente che esegue il servizio IBM MQ.
 - i) Fare clic su **Avvia > Esegui ...**, immettere `services.msc` e fare clic su **OK**.
 - ii) Fare clic sul servizio IBM MQ nell'elenco di servizi e fare clic sulla scheda Accesso.
 - iii) Ricordare l'ID utente e chiudere Esplora servizi.
- e) Aggiungere l'ID utente che esegue il servizio IBM MQ al gruppo *al_tmqm*. Aggiungere inoltre l'ID utente con cui si accede per creare un gestore code ed eseguirlo in modo interattivo.

Windows controlla l'autorizzazione del gestore code per accedere alle directory di dati e log controllando l'autorizzazione dell'ID utente che sta eseguendo i processi del gestore code. L'ID utente deve essere un membro, direttamente o indirettamente tramite un gruppo globale, del gruppo *al_tmqm* che ha autorizzato le directory.

Se IBM MQ è stato installato come parte di un dominio e verranno eseguite le attività in [“Creazione di un gestore code a più istanze su workstation o server di dominio su Windows” a pagina 529](#), gli ID utente del dominio creati in [“Creazione di un dominio Active Directory e DNS su Windows” a pagina 532](#) sono *wmquser1* e *wmquser2*.

Se il gestore code non è stato installato come parte di un dominio, l'ID utente locale predefinito che esegue il servizio IBM MQ è `MUSR_MQADMIN`. Se si intende eseguire le attività senza l'autorizzazione di amministratore, creare un utente che sia un membro del gruppo `mqm` locale. Seguire questi passi per aggiungere *wmquser1* e *wmquser2* a *al_tmqm*. Se la configurazione è diversa, sostituire i nomi per gli ID utente e il gruppo.

- i) Nell'elenco dei gruppi, fare clic con il tasto destro del mouse su **al_tmqm > Proprietà > Aggiungi ...**
- ii) Nella finestra Seleziona utenti, computer o gruppi, immettere *wmquser1* ; *wmquser2* e fare clic su **Controlla nomi**.
- iii) Immettere il nome e la parola d'ordine di un amministratore di dominio nella finestra Windows Sicurezza, quindi fare clic su **OK > OK > Applica > OK**.

2. Aprire un prompt dei comandi.

3. Riavviare il servizio IBM MQ.

È necessario riavviare il servizio in modo che l'ID utente con cui viene eseguito acquisisca le ulteriori credenziali di sicurezza configurate per esso.

Immettere i comandi:

```
endmqsvc  
strmqsvc
```

Le risposte del sistema:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

E:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. Immettere il comando:

```
md c:\wmq\data, c:\wmq\logs
```

5. Impostare le autorizzazioni sulle directory per consentire all'utente locale *user* l'accesso in lettura e scrittura.

```
cacls c:\wmq/T /E /G almqm:F
```

La risposta del sistema:

```
processed dir: c:\wmq  
processed dir: c:\wmq\data  
processed dir: c:\wmq\logs
```

6. Opzionale: Passare a un ID utente che è un membro del gruppo *mqm* locale.

È possibile continuare come amministratore, ma per una configurazione di produzione realistica, continuare con un ID utente con diritti più limitati. L'ID utente deve essere almeno un membro del gruppo *mqm* locale.

Se l'installazione di IBM MQ è configurata come parte di un dominio, rendere l'ID utente un membro del gruppo *Domain mqm*. La procedura guidata "Prepara IBM MQ" rende il gruppo globale *Domain mqm* un membro del gruppo *mqm* locale, quindi non è necessario rendere l'ID utente direttamente un membro del gruppo *mqm* locale.

7. Creare il gestore code.

```
crtmqm -a almqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

La risposta del sistema:

```
IBM MQ queue manager created.  
Directory 'c:\wmq1\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

8. Verificare che le directory create dal gestore code si trovino nella directory *c:\wmq*.

```
dir c:\wmq\D /B /S
```

9. Verificare che i file dispongano dell'autorizzazione di lettura e scrittura o di controllo completo per il gruppo mqm locale.

```
cacls c:\wmq\*.*
```

Operazioni successive

Verificare il gestore code inserendo e ricevendo un messaggio in una coda.

1. Avviare il gestore code.

```
strmqm QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. Creare una coda di test.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

La risposta del sistema:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. Inserire un messaggio di verifica utilizzando il programma di esempio **amqsput**.

```
echo 'A test message' | amqsput QTEST QMGR
```

La risposta del sistema:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. Richiamare il messaggio di test utilizzando il programma di esempio **amqsget**.

```
amqsget QTEST QMGR
```

La risposta del sistema:

```
Sample AMQSGETO start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGETO end
```

5. Chiudere il gestore code.

```
endmqm -i QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. Eliminare il gestore code.

```
dltmqm QMGR
```

La risposta del sistema:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. Eliminare le directory create.

Suggerimento: Aggiungere l'opzione /Q ai comandi per evitare che il comando richieda di eliminare ogni file o directory.

```
del /F /S C:\wmq\*. *
rmdir /S C:\wmq
```

Attività correlate

Windows [Lettura e scrittura di dati e file di log autorizzati dal gruppo mqm locale](#)

L'attività illustra come creare un gestore code con i relativi dati e file di log memorizzati in qualsiasi directory di propria scelta. L'accesso ai file è protetto dal gruppo mqm locale. La directory non è condivisa.

Linux [Creazione di un gestore code a più istanze su Linux](#)

Un esempio che mostra come configurare un gestore code a più istanze su Linux. L'impostazione è piccola per illustrare i concetti coinvolti. L'esempio si basa su Linux Red Hat Enterprise 5. I passi differiscono su altre piattaforme UNIX .

Informazioni su questa attività

L'esempio è configurato su un computer notebook da 2 GHz con 3 GB di RAM su cui è in esecuzione Windows 7 Service Pack 1. Due macchine virtuali VMware , Server1 e Server2, eseguono Linux Red Hat Enterprise 5 in immagini da 640 MB. Server1 ospita il file system di rete (NFS), i log del gestore code e un'istanza HA. Non è consuetudine per il server NFS ospitare anche una delle istanze del gestore code; questo per semplificare l'esempio. Server2 monta i log del gestore code del Server1 con un'istanza in standby. Un client MQI WebSphere MQ viene installato su un'immagine VMware aggiuntiva di 400 MB che esegue Windows 7 Service Pack 1 ed esegue le applicazioni ad alta disponibilità di esempio. Tutte le macchine virtuali sono configurate come parte di una rete solo host VMware per ragioni di sicurezza.

Nota: È necessario inserire solo i dati del gestore code su un server NFS . Su NFS, utilizzare le seguenti tre opzioni con il comando mount per rendere il sistema sicuro:

- **noexec**
Utilizzando questa opzione, si impedisce l'esecuzione dei file binari su NFS, il che impedisce a un utente remoto di eseguire codice indesiderato sul sistema.
- **nosuid**
Utilizzando questa opzione, si impedisce l'utilizzo dei bit set - user - identifier e set - group - identifier, che impediscono a un utente remoto di ottenere privilegi più elevati.
- **nessun dev**
Utilizzando questa opzione, si arrestano i caratteri e si bloccano i dispositivi speciali da utilizzare o definire, il che impedisce a un utente remoto di uscire da una prigione chroot.

Procedura

1. Collegarsi come root.
2. Leggere la sezione [Installazione IBM MQ - panoramica](#) e seguire il collegamento appropriato per installare IBM MQ, creare l'utente e il gruppo mqm e definire /var/mqm.
3. Completare l'attività [Verifica del funzionamento del file system condiviso](#) per verificare che il file system supporti gestori code a più istanze.
4. Per Server1, completare la seguente procedura:
 - a. Creare le directory di log e di dati in una cartella comune, /MQHA, da condividere. Ad esempio:
 - i) **mkdir /MQHA**
 - ii) **mkdir /MQHA/logs**
 - iii) **mkdir /MQHA/qmgrs**
5. Per Server2, completare il seguente passo:
 - a. Creare la cartella, /MQHA, per montare il file system condiviso. Mantenere lo stesso percorso di Server1. Ad esempio:
 - i) **mkdir /MQHA**
6. Assicurarsi che le directory MQHA siano di proprietà dell'utente e del gruppo mqm e che le autorizzazioni di accesso siano impostate su rwx per utente e gruppo. Ad esempio, **ls -al** visualizza drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA.
 - a. **chown -R mqm:mqm /MQHA**
 - b. **chmod -R ug+rwx /MQHA**
7. Creare il gestore code immettendo il seguente comando: **crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1**
8. Aggiungere²Da /MQHA *(rw, sync, no_wdelay, fsid=0) a /etc/exports
9. Per Server1, completare la seguente procedura:
 - a. Avviare il daemon NFS : **/etc/init.d/ nfs start**
 - b. Copiare i dettagli di configurazione del gestore code da Server1: :

```
dspmqlnf -o command QM1
```

e copiare il risultato negli appunti:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

² '*' consente a tutte le macchine che possono raggiungere questo montaggio /MQHA per la lettura/scrittura. Limitare l'accesso su una macchina di produzione.

10. Per Server2, completare la seguente procedura:

- a. Montare il file system esportato /MQHA immettendo il seguente comando: **mount -t nfs4 -o hard,intr Server1:/ /MQHA**
- b. Incollare il comando di configurazione del gestore code in Server2:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

11. Avviare le istanze del gestore code, in entrambi gli ordini, con il parametro **-x : stmqm -x QM1**.

Il comando utilizzato per avviare le istanze del gestore code deve essere immesso dalla stessa installazione IBM MQ del comando **addmqinf** . Per avviare e arrestare il gestore code da una installazione diversa, è necessario prima impostare l'installazione associata al gestore code utilizzando il comando **setmqm** . Per ulteriori informazioni, vedere [setmqm](#).

Linux *Verifica del gestore code a più istanze su Linux*

Utilizzare i programmi di esempio **amqsgshac**, **amqspshac** e **amqsmhac** per verificare la configurazione di un gestore code a più istanze. Questo argomento fornisce una configurazione di esempio per verificare una configurazione del gestore code a più istanze su Linux Red Hat Enterprise 5.

I programmi di esempio ad alta disponibilità utilizzano la riconnessione client automatica. Quando il gestore code connesso ha esito negativo, il client tenta di riconnettersi a un gestore code nello stesso gruppo di gestori code. La descrizione degli esempi, [Programmi di esempio ad alta disponibilità](#), illustra la riconnessione del client utilizzando un gestore code a istanza singola per semplicità. È possibile utilizzare gli stessi esempi con i gestori code a più istanze per verificare una configurazione del gestore code a più istanze.

L'esempio utilizza la configurazione a più istanze descritta in ["Creazione di un gestore code a più istanze su Linux"](#) a pagina 566. Utilizzare la configurazione per verificare che il gestore code a più istanze passi all'istanza in standby. Arrestare il gestore code con il comando **endmqm** e utilizzare l'opzione **-s**, **switchover**. I programmi client si riconnettono alla nuova istanza del gestore code e continuano a lavorare con la nuova istanza dopo un leggero ritardo.

Nell'esempio, il client è in esecuzione su Windows 7 Service Pack 1. Il sistema ospita due server VMware Linux che eseguono il gestore code a più istanze.

Verifica del failover utilizzando IBM MQ Explorer

Prima di utilizzare le applicazioni di esempio per verificare il failover, eseguire IBM MQ Explorer su ciascun server. Aggiungere entrambe le istanze del gestore code a ciascun explorer utilizzando la procedura guidata **Aggiungi gestore code remoto > Connetti direttamente a gestore code a più istanze** . Verificare che entrambe le istanze siano in esecuzione, consentendo lo standby. Chiudere la finestra che esegue l'immagine VMware con l'istanza attiva, spegnendo virtualmente il server o arrestando l'istanza attiva, consentendo il passaggio all'istanza standby.

Nota: Se si spegne il server, assicurarsi che non sia quello che ospita /MQHA !

Nota: L'opzione **Consenti commutazione a un'istanza in standby** potrebbe non essere disponibile nella finestra di dialogo **Arresta gestore code** . L'opzione è mancante perché il gestore code è in esecuzione come gestore code a istanza singola. È necessario che sia stato avviato senza l'opzione **Consenti un'istanza standby** . Se la richiesta di arresto del gestore code viene rifiutata, consultare la finestra **Dettagli** , probabilmente perché non è in esecuzione alcuna istanza in standby.

Verifica del failover utilizzando i programmi di esempio

Scegliere un server da utilizzare per eseguire l'istanza attiva

È possibile che sia stato scelto uno dei server per ospitare la directory o il file system MQHA . Se si prevede di verificare il failover chiudendo la finestra VMware che esegue il server attivo, assicurarsi che non sia quello che ospita MQHA !

Sul server che esegue l'istanza del gestore code attivo

Nota: L'esecuzione del canale SVRCONN con MCAUSER impostato su mqm, è una comodità per ridurre il numero di fasi di configurazione nell'esempio. Se viene scelto un altro ID utente e il proprio sistema è impostato in modo diverso da quello utilizzato nell'esempio, è possibile che si verifichino problemi di autorizzazione di accesso. Non utilizzare mqm come MCAUSER su un sistema esposto; è probabile che comprometta notevolmente la sicurezza.

1. Modificare *ipaddr1* e *ipaddr2* e salvare i seguenti comandi in */MQHA/hasamples.tst*.

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2
(1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. Aprire una finestra di terminale con il percorso */MQHA* ed eseguire il comando:

```
runmqsc -m QM1 < hasamples.tst
```

3. Verificare che il listener sia in esecuzione e che disponga del controllo del gestore code, ispezionando l'output del comando **runmqsc**.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

Oppure, utilizzando il IBM MQ Explorer che il listener TCP/IP è in esecuzione e dispone di Control = Queue Manager.

Sul client

1. Copiare la tabella di connessione client AMQCLCHL.TAB da */MQHA/qmgrs/QM1.000/@ipcc* sul server a *C:* sul client.
2. Aprire un prompt dei comandi con il percorso *C:* e impostare la variabile di ambiente MQCHLLIB in modo che punti alla tabella di definizione del canale client (CCDT)

```
SET MQCHLLIB=C:\
```

3. Al prompt dei comandi immettere i seguenti comandi:

```
start amqsgnac TARGET QM1
start amqsmnac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

Sul server che esegue l'istanza del gestore code attivo

1. Le alternative sono:

- Chiudere la finestra che esegue l'immagine VMware con l'istanza del server attiva.

- Utilizzando IBM MQ Explorer, arrestare l'istanza del gestore code attivo, consentendo la commutazione all'istanza in standby e istruendo i client riconnettibili a riconnettersi.
2. I tre client alla fine rilevano che la connessione è interrotta e quindi si ricollegano. In questa configurazione, se si chiude la finestra del server, sono necessari circa sette minuti per ristabilire tutte e tre le connessioni. Alcune connessioni vengono ristabilite ben prima di altre.

Risultati

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

Conversione di una singola istanza in un gestore code a più istanze su AIX and Linux

Per convertire un gestore code a istanza singola in un gestore code a più istanze su AIX and Linux, è necessario spostare i dati del gestore code in una directory condivisa e riconfigurare il gestore code su due altri server.

Prima di iniziare

È necessario controllare i prerequisiti per eseguire un gestore code a più istanze come parte di questa attività. Per un elenco di ambienti verificati, fare riferimento a [Verifica dell'istruzione per i file system del gestore code a più istanze IBM MQ](#). Altri ambienti potrebbero funzionare; uno strumento di test viene fornito con IBM MQ per assistere l'utente nella qualificazione di altri ambienti.

È necessario disporre di tre server per eseguire un gestore code a più istanze. Un server ha un file system condiviso per memorizzare i log e i dati del gestore code. Gli altri server eseguono le istanze attive e in standby del gestore code.

Informazioni su questa attività

Si dispone di un gestore code a istanza singola che si desidera convertire in gestore code a più istanze. La conversione del gestore code è semplice, ma è necessario eseguire altre attività per creare un ambiente di produzione completamente automatizzato.

È necessario controllare i prerequisiti per un gestore code a più istanze, configurare l'ambiente e controllarlo. È necessario impostare un sistema di monitoraggio e di gestione per rilevare se il gestore code a più istanze ha avuto esito negativo ed è stato riavviato automaticamente. È quindi possibile individuare la causa del riavvio, risolvere il problema e riavviare lo standby. Inoltre, è necessario modificare le applicazioni o il modo in cui le applicazioni sono connesse al gestore code, in modo che possano riprendere l'elaborazione dopo il riavvio di un gestore code.

Procedura

1. Controllare il sistema operativo su cui si eseguirà il gestore code e il file system su cui sono memorizzati i dati e i log del gestore code. Verificare che possano eseguire un gestore code a più istanze.
 - a) Consultare [Verifica dell'istruzione per i file system del gestore code a più istanze IBM MQ](#). Verificare se la combinazione di sistema operativo e file system è verificata e in grado di eseguire un gestore code a più istanze.

Un file system condiviso deve fornire un blocco basato sul lease per essere adeguato per eseguire gestori code a più istanze. Il blocco basato sul lease è una funzione recente di alcuni file system condivisi e in alcuni casi sono necessarie correzioni. La dichiarazione di supporto fornisce le informazioni essenziali.
 - b) Eseguire **amqmfscck** per verificare che il file system sia configurato correttamente.

I file system sono a volte configurati con prestazioni superiori rispetto all'integrità dei dati. È importante controllare la configurazione del file system. Un report negativo dello strumento **amqmfscck** indica che le impostazioni non sono adeguate. Un risultato positivo indica che il file system è adeguato, ma il risultato non è un'istruzione definitiva che il file system sia adeguato. È una buona indicazione.
 - c) Eseguire l'applicazione di controllo di integrità fornita nella nota tecnica, [Verifica di un file system condiviso per la compatibilità con IBM MQ Gestori code a più istanze](#).

Il controllo dell'applicazione verifica che il gestore code venga riavviato correttamente.
2. Configurare un utente e un gruppo per poter accedere a una condivisione sul filesystem di rete da ogni server che esegue un'istanza del gestore code.

Su AIX and Linux, `uid` e il `gid` per `mqm` in `/etc/passwd` devono essere gli stessi su ciascun sistema; consultare [Creazione di un gestore code a più istanze su Linux](#).
3. Impostare una directory per la condivisione sul file system di rete con le autorizzazioni di accesso corrette.

Una configurazione tipica è quella di impostare una singola directory condivisa che contiene tutte le directory di dati e di log per tutti i gestori code che utilizzano il disco condiviso; vedere [Condividi qmgrs e directory di log in Configurazioni di directory di esempio sui sistemi AIX and Linux](#).

Ad esempio, creare una directory root sulla condivisione denominata MQHA che ha sottodirectory `data` e `logs`. Ogni gestore code crea le proprie directory di dati e di log in `data` e `logs`. Creare /MQHA sull'unità condivisa. /MQHA è di proprietà dell'utente e del gruppo `mqm` e dispone delle autorizzazioni di accesso `rwx`.
4. Copiare i dati del gestore code e i log nella condivisione.

Seguire la procedura per eseguire il backup del gestore code descritta in [Backup dei dati del gestore code](#).

Nota: Diversamente da Windows, il programma di utilità hamvmqm non può essere utilizzato su AIX and Linux.

5. Aggiornare le informazioni di configurazione del gestore code memorizzate sul server del gestore code corrente completando la seguente procedura:

- a) Modificare la stanza Log: nel file `qm.ini` del gestore code, che si trova in *share*:

```
LogPath= share/logs/QMgrName
```

- b) Modificare la sezione QueueManager: del file IBM MQ `mqc.ini`, che si trova generalmente nella directory `/var/mqm` su AIX and Linux:

```
DataPath= share/data/QMgrName
```

dove *QMgrName* è il nome Directory nella stanza QueueManager: nel file `mqc.ini` e *share* è la condivisione dove vengono spostati i dati e i log.

6. Aggiungere le informazioni di configurazione del gestore code al nuovo server del gestore code.

- a) Eseguire il comando **dspmqlinf** per visualizzare le informazioni sul gestore code.

Eseguire il comando sul server che ha eseguito il gestore code.

```
dspmqlinf -o command QMgrName
```

L'output del comando è formattato per creare una configurazione del gestore code.

```
addmqinf -s QueueManager -v Name= QMgrName -v Directory= QMgrName -v  
Prefix=d:\var\mqm Datapath= \share\data\QMgrName
```

- b) Creare una configurazione del gestore code sull'altro server.
Eseguire il comando **addmqinf** copiato dall'output precedente.

7. Aggiungere l'indirizzo di rete del nuovo server al nome connessione nelle definizioni di client e canale.

- a) Trovare tutte le impostazioni TCPIP del client, del mittente e del richiedente che fanno riferimento al server.

Le impostazioni client potrebbero essere in CCDT (Client Definition Tables), in variabili di ambiente, in file delle proprietà Java o nel codice client. I canali cluster rilevano automaticamente il nome connessione di un gestore code dal relativo canale ricevente del cluster. Finché il nome del canale ricevente del cluster è vuoto o omissso, TCPIP rileva l'indirizzo IP del server che ospita il gestore code.

- b) Modificare il nome connessione per ognuna di queste connessioni per includere gli indirizzi TCPIP di entrambi i server che ospitano il gestore code a più istanze.

Ad esempio, modificare il seguente nome connessione:

```
echo DISPLAY CHANNEL(ENGLAND) CONNAME | runmqsc QM1
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QM1.  
1: DISPLAY CHANNEL(ENGLAND) CONNAME  
AMQ8414: Display Channel details.  
CHANNEL(ENGLAND) CHLTYPE(SDR)  
CONNAME(LONDON)
```

in:

```
echo ALTER CHANNEL(ENGLAND) CHLTYPE(SDR) CONNAME('LONDON, BRISTOL') | runmqsc QM1
```

8. Aggiornare le procedure di controllo e gestione per rilevare il riavvio del gestore code.
9. Aggiornare le applicazioni client in modo che siano ricollegabili automaticamente, se appropriato.
10. Aggiornare la procedura di avvio per le applicazioni IBM MQ da avviare come servizi del gestore code.
11. Avviare ogni istanza del gestore code, consentendone l'elevata disponibilità.

La prima istanza del gestore code avviata diventa l'istanza attiva. Emettere il comando due volte, una su ciascun server.

```
strmqm -x QMgrName
```

Operazioni successive

Per ottenere la massima disponibilità dai gestori code a più istanze, è necessario progettare le applicazioni client in modo che siano ricollegabili e le applicazioni server riavviabili; consultare [Ripristino applicazione](#).

Concetti correlati

[Ripristino applicazione](#)

[Riconnessione automatica del client](#)

[Riconnessione canale e client](#)

[Gestori code a più istanze](#)

[File di configurazione del gestore code, qm.ini](#)

[File system condiviso](#)

Attività correlate

[Backup dei dati del gestore code](#)

[Modifica delle informazioni di configurazione IBM MQ su Multiplatforms](#)

[Creazione di un gestore code a più istanze su Linux](#)

[Spostamento di un gestore code nella memoria MSCS](#)

[Verifica del blocco del file system condiviso](#)

Riferimenti correlati

amqmfsc (controllo file system)

[File di configurazione IBM MQ , mqs.ini](#)

Informazioni correlate

[Verifica di un file system condiviso per la compatibilità con i gestori code a più istanze IBM MQ](#)

[Istruzione di verifica per file system di gestori code a più istanze IBM MQ](#)

Windows

Conversione di una singola istanza in un gestore code a più istanze su Windows

Per convertire un singolo gestore code di istanza, in un gestore code a più istanze, su piattaforme Windows , è necessario spostare i dati del gestore code in una directory condivisa e riconfigurare il gestore code su due altri server.

Prima di iniziare

È necessario controllare i prerequisiti per l'esecuzione di un gestore code a più istanze come parte di questa attività. Per un elenco di ambienti verificati, consultare [Verifica dell'istruzione per i file system del gestore code a più istanze IBM MQ](#). Altri ambienti potrebbero funzionare; uno strumento di test viene fornito con IBM MQ per assistere l'utente nella qualificazione di altri ambienti.

È necessario disporre di tre server per eseguire un gestore code a più istanze. Un server ha un file system condiviso per memorizzare i log e i dati del gestore code. Gli altri server eseguono le istanze attive e in standby del gestore code.

Informazioni su questa attività

Si dispone di un gestore code a istanza singola che si desidera convertire in gestore code a più istanze. La conversione del gestore code è semplice, ma è necessario eseguire altre attività per creare un ambiente di produzione completamente automatizzato.

È necessario controllare i prerequisiti per un gestore code a più istanze, configurare l'ambiente e controllarlo. È necessario impostare un sistema di monitoraggio e di gestione per rilevare se il gestore

code a più istanze ha avuto esito negativo ed è stato riavviato automaticamente. È quindi possibile individuare la causa del riavvio, risolvere il problema e riavviare lo standby. Inoltre, è necessario modificare le applicazioni o il modo in cui le applicazioni sono connesse al gestore code, in modo che possano riprendere l'elaborazione dopo il riavvio di un gestore code.

Procedura

1. Controllare il sistema operativo su cui si eseguirà il gestore code e il file system su cui sono memorizzati i dati e i log del gestore code. Verificare che possano eseguire un gestore code a più istanze.
 - a) Consultare [Verifica dell'istruzione per i file system del gestore code a più istanze IBM MQ](#). Verificare se la combinazione di sistema operativo e file system è verificata e in grado di eseguire un gestore code a più istanze.

Un file system condiviso deve fornire un blocco basato sul lease per essere adeguato per eseguire gestori code a più istanze. Il blocco basato sul lease è una funzione recente di alcuni file system condivisi e in alcuni casi sono necessarie correzioni. La dichiarazione di supporto fornisce le informazioni essenziali.
 - b) Eseguire l'applicazione di controllo di integrità fornita nella nota tecnica, [Verifica di un file system condiviso per la compatibilità con IBM MQ Gestori code a più istanze](#).

Il controllo dell'applicazione verifica che il gestore code venga riavviato correttamente.
2. Configurare un utente e un gruppo per poter accedere a una condivisione sul filesystem di rete da ogni server che esegue un'istanza del gestore code.

Su Windows, gli ID di sicurezza (SID) del gruppo mqm possono essere diversi; fare riferimento a [Windows domini e gestori code a più istanze](#).
3. Impostare una directory per la condivisione sul file system di rete con le autorizzazioni di accesso corrette.

Una configurazione tipica consiste nell'impostare una singola directory condivisa che contiene tutte le directory di dati e di log per tutti i gestori code che utilizzano il disco condiviso; consultare [Condividi qmgrs e directory di log denominate](#).

Ad esempio, creare una directory root sulla condivisione denominata MQHA che ha sottodirectory data e logs. Ogni gestore code crea le proprie directory di dati e di log in data e logs. Creare *drive* \MQHA sull'unità condivisa. Il proprietario è membro di mqm. mqm deve avere l'autorizzazione di controllo completo. Creare una condivisione per *drive*\MQHA.
4. Copiare i dati del gestore code e i log nella condivisione.

Su Windows, è possibile eseguire il comando **hamvmqm** per spostare i dati del gestore code nella condivisione.
5. Aggiungere le informazioni di configurazione del gestore code al nuovo server del gestore code.
 - a) Eseguire il comando **dspmqinf** per visualizzare le informazioni sul gestore code
Eseguire il comando sul server che ha eseguito il gestore code.

```
dspmqinf -o command QMgrName
```

L'output del comando è formattato per creare una configurazione del gestore code.

```
addmqinf -s QueueManager -v Name= QMgrName -v Directory= QMgrName -v  
Prefix=d:\var\mqm Datapath= \share\data\QMgrName
```

 - b) Creare una configurazione del gestore code sull'altro server.
Eseguire il comando **addmqinf** copiato dall'output precedente.
6. Aggiungere l'indirizzo di rete del nuovo server al nome connessione nelle definizioni di client e canale.
 - a) Trovare tutte le impostazioni TCPIP del client, del mittente e del richiedente che fanno riferimento al server.
 - Le impostazioni client potrebbero essere in CCDT (Client Definition Tables), in variabili di ambiente, in file delle proprietà Java o nel codice client.

- I canali cluster rilevano automaticamente il nome connessione di un gestore code dal relativo canale ricevente del cluster. Finché il nome del canale ricevente del cluster è vuoto o omissivo, TCP/IP rileva l'indirizzo IP del server che ospita il gestore code.
- b) Modificare il nome connessione per ognuna di queste connessioni per includere gli indirizzi TCP/IP di entrambi i server che ospitano il gestore code a più istanze.
Ad esempio, modificare il seguente nome connessione:

```
echo DISPLAY CHANNEL(ENGLAND) CONNAME | runmqsc QM1
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
1: DISPLAY CHANNEL(ENGLAND) CONNAME
AMQ8414: Display Channel details.
CHANNEL(ENGLAND) CHLTYPE(SDR)
CONNAME(LONDON)
```

in:

```
echo ALTER CHANNEL(ENGLAND) CHLTYPE(SDR) CONNAME('LONDON, BRISTOL') | runmqsc QM1
```

7. Aggiornare le procedure di controllo e gestione per rilevare il riavvio del gestore code.
8. Aggiornare le applicazioni client in modo che siano ricollegabili automaticamente, se appropriato.
9. Aggiornare la procedura di avvio per le applicazioni IBM MQ da avviare come servizi del gestore code.
10. Avviare ogni istanza del gestore code, consentendone l'elevata disponibilità.

La prima istanza del gestore code avviata diventa l'istanza attiva. Emettere il comando due volte, una su ciascun server.

```
strmqm -x QMgrName
```

Operazioni successive

Per ottenere la massima disponibilità dai gestori code a più istanze, è necessario progettare le applicazioni client in modo che siano ricollegabili e le applicazioni server riavviabili; consultare [Ripristino applicazione](#).

Concetti correlati

[Ripristino applicazione](#)

[Riconnessione automatica del client](#)

[Riconnessione canale e client](#)

[Gestori code a più istanze](#)

[File di configurazione del gestore code, qm.ini](#)

[File system condiviso](#)

[Domini Windows e gestori code a più istanze](#)

Attività correlate

[Backup dei dati del gestore code](#)

[Modifica delle informazioni di configurazione IBM MQ su Multiplatforms](#)

[Spostamento di un gestore code nella memoria MSCS](#)

[Verifica del blocco del file system condiviso](#)

[Gestione dei servizi](#)

Riferimenti correlati

[amqmfsc \(controllo file system\)](#)

Informazioni correlate

[Verifica di un file system condiviso per la compatibilità con i gestori code a più istanze IBM MQ](#)

Multi

Eliminazione di un gestore code a più istanze

Su Multiplatforms, per eliminare completamente un gestore code a più istanze, utilizzare il comando **dltmqm** per eliminare il gestore code e rimuovere le istanze da altri server utilizzando i comandi **rmvmqinf** o **dltmqm**.

Eseguire il comando **dltmqm** per eliminare un gestore code con istanze definite su altri server, su qualsiasi server in cui è definito tale gestore code. Non è necessario eseguire il comando **dltmqm** sullo stesso server su cui è stato creato. Quindi, eseguire il comando **rmvmqinf** o **dltmqm** su tutti gli altri server che hanno una definizione del gestore code.

È possibile eliminare un gestore code solo quando è arrestato. Nel momento in cui si elimina, non ci sono istanze in esecuzione e il gestore code, in senso stretto, non è né un gestore code a istanza singola né un gestore code a più istanze; è semplicemente un gestore code che ha i dati del gestore code e i log su una condivisione remota. Quando si elimina un gestore code, i relativi log e dati del gestore code vengono eliminati e la stanza del gestore code viene rimossa dal file `mqs.ini` sul server su cui è stato immesso il comando **dltmqm**. È necessario disporre dell'accesso alla condivisione di rete contenente i dati e i log del gestore code quando si elimina il gestore code.

Su altri server in cui sono state precedentemente create istanze del gestore code, sono presenti anche voci nei file `mqs.ini` su tali server. È necessario visitare ciascun server e rimuovere la sezione del gestore code eseguendo il comando **rmvmqinf** *Nome stanza gestore code*.

Linux

AIX

Su sistemi AIX and Linux, se è stato inserito un file `mqs.ini` comune nella memoria di rete e si fa riferimento ad esso da tutti i server impostando la variabile di ambiente `AMQ_MQS_INI_LOCATION` su ciascun server, è necessario eliminare il gestore code da uno solo dei relativi server poiché esiste un solo file `mqs.ini` da aggiornare.

Esempio

Primo server

```
dltmqm QM1
```

Altri server in cui sono definite le istanze

```
rmvmqinf QM1, oppure
```

```
dltmqm QM1
```

Linux

AIX

Ripristino a un gestore code a istanza singola su AIX and Linux

Ripristinare un gestore code a più istanze a un gestore code a istanza singola, su AIX and Linux, arrestando l'istanza in standby. Quindi riavviare l'istanza attiva e non impostare l'indicatore che consente le istanze in standby.

Prima di iniziare

Sono presenti almeno tre server configurati per eseguire un gestore code come gestore code a più istanze. Il gestore code è attualmente in esecuzione come un gestore code a più istanze, con un'istanza in standby attiva.

Informazioni su questa attività

L'attività implica la disattivazione dello standby attivo in modo che rimanga attivo solo il gestore code a più istanze in esecuzione. Per evitare che un'istanza in standby venga avviata in futuro, è necessario arrestare l'istanza attiva e riavviarla. Quando lo si riavvia, lo si avvia come un gestore code a istanza singola che impedisce l'avvio delle istanze in standby. L'istanza standby viene arrestata come passo separato, per darti la possibilità di riavviare l'istanza attiva in un secondo momento. È possibile arrestare entrambe le istanze eseguendo il comando `endmqm QMgrName standard` sul server che esegue il gestore code attivo.

Procedura

1. Arrestare l'istanza del gestore code in standby.

Sul server su cui è in esecuzione l'istanza standby:

```
endmqm -w QMgrName
```

2. Arrestare l'istanza del gestore code attivo.

Sul server che esegue l'istanza attiva:

```
endmqm -w (QMgrName)
```

3. Riavviare il gestore code, impedendo gli standby.

Sul server che eseguirà il gestore code:

```
strmqm QMgrName
```

Operazioni successive

È possibile eseguire il gestore code come una singola istanza sullo stesso server dei dati del gestore code.

Quando il gestore code viene arrestato, spostare nuovamente i dati del gestore code sul server su cui è in esecuzione il gestore code. In alternativa, installare IBM MQe spostare la definizione di configurazione del gestore code sul server con i dati del gestore code. Entrambe le attività sono variazioni dei passi in [“Conversione di una singola istanza in un gestore code a più istanze su AIX and Linux” a pagina 570](#) per creare un gestore code a più istanze.

Windows *Ripristino a un gestore code a istanza singola su Windows*

Ripristinare un gestore code a più istanze a un gestore code a singola istanza, su piattaforme Windows , arrestando l'istanza in standby. Quindi riavviare l'istanza attiva e non impostare l'indicatore che consente le istanze in standby.

Prima di iniziare

Sono presenti almeno tre server configurati per eseguire un gestore code come gestore code a più istanze. Il gestore code è attualmente in esecuzione come un gestore code a più istanze, con un'istanza in standby attiva.

Informazioni su questa attività

L'attività implica la disattivazione dello standby attivo in modo che rimanga attivo solo il gestore code a più istanze in esecuzione. Per evitare che un'istanza in standby venga avviata in futuro, è necessario arrestare l'istanza attiva e riavviarla. Quando lo si riavvia, lo si avvia come un gestore code a istanza singola che impedisce l'avvio delle istanze in standby. L'istanza standby viene arrestata come passo separato, per darti la possibilità di riavviare l'istanza attiva in un secondo momento. È possibile arrestare entrambe le istanze eseguendo il comando `endmqm QMgrName standard` sul server che esegue il gestore code attivo.

Procedura

1. Arrestare l'istanza del gestore code in standby.

Sul server su cui è in esecuzione l'istanza standby:

```
endmqm -w QMgrName
```

2. Arrestare l'istanza del gestore code attivo.

Sul server che esegue l'istanza attiva:

```
endmqm -w (QMgrName)
```

3. Riavviare il gestore code, impedendo gli standby.

Sul server che eseguirà il gestore code:

```
strmqm QMgrName
```

Operazioni successive

È possibile eseguire il gestore code come una singola istanza sullo stesso server dei dati del gestore code.

Quando il gestore code viene arrestato, spostare nuovamente i dati del gestore code sul server su cui è in esecuzione il gestore code. In alternativa, installare IBM MQ e spostare la definizione di configurazione del gestore code sul server con i dati del gestore code. Entrambe le attività sono variazioni di passi in [Conversione di una singola istanza in un gestore code a più istanze su Windows](#).

Avvio e arresto di un gestore code a più istanze

Avvio e arresto di un gestore code configurato su Multiplatforms come una singola istanza o un gestore code a più istanze.

Una volta definito un gestore code a più istanze su una coppia di server, è possibile eseguire il gestore code su entrambi i server, come gestore code a istanza singola o come gestore code a più istanze.

Per eseguire un gestore code a più istanze, avviare il gestore code su uno dei server utilizzando il comando **strmqm -x QM1** ; l'opzione -x consente il failover dell'istanza. Diventa l' *istanza attiva*. Avviare l'istanza di standby sull'altro server utilizzando lo stesso comando **strmqm -x QM1** ; l'opzione -x consente l'avvio dell'istanza come standby.

Il gestore code è ora in esecuzione con un'istanza attiva che sta elaborando tutte le richieste e un'istanza in standby che è pronta a subentrare se l'istanza attiva ha esito negativo. All'istanza attiva viene concesso l'accesso esclusivo ai log e ai dati del gestore code. Lo standby attende che venga concesso l'accesso esclusivo ai log e ai dati del gestore code. Quando lo standby ha accesso esclusivo, diventa l'istanza attiva.

È anche possibile passare manualmente il controllo all'istanza standby immettendo il comando **endmqm -s** sull'istanza attiva. Il comando **endmqm -s** arresta l'istanza attiva senza arrestare lo standby. Il blocco di accesso esclusivo sui log e sui dati del gestore code viene rilasciato e lo standby assume il controllo.

È anche possibile avviare e arrestare un gestore code configurato con più istanze su server diversi come gestore code a istanza singola. Se si avvia il gestore code senza utilizzare l'opzione -x nel comando **strmqm** , alle istanze del gestore code configurato su altre macchine viene impedito l'avvio come istanze in standby. Se si tenta di avviare un'altra istanza, si riceve la risposta che l'istanza del gestore code non può essere eseguita come standby.

Se si arresta l'istanza attiva di un gestore code a più istanze utilizzando il comando **endmqm** senza l'opzione -s , le istanze attive e in standby si arrestano entrambe. Se si arresta l'istanza in standby utilizzando il comando **endmqm** con l'opzione -x , smette di essere in standby e l'istanza attiva continua l'esecuzione. Non è possibile emettere **endmqm** senza l'opzione -x sullo standby.

Solo due istanze del gestore code possono essere eseguite contemporaneamente; una è l'istanza attiva e l'altra è un'istanza in standby. Se si avviano due istanze contemporaneamente, IBM MQ non ha alcun controllo su quale istanza diventa attiva; è determinata dal file system di rete. La prima istanza che acquisisce l'accesso esclusivo ai dati del gestore code diventa l'istanza attiva.

Nota: Prima di riavviare un gestore code non riuscito, è necessario disconnettere le applicazioni da tale istanza del gestore code. In caso contrario, il gestore code potrebbe non essere riavviato correttamente.

File system condiviso

Su piattaforme multiple, un gestore code a più istanze utilizza un filesystem di rete per gestire le istanze del gestore code.

Un gestore code a più istanze automatizza il failover utilizzando una combinazione di blocchi del filesystem e log e dati del gestore code condivisi. Solo una istanza di un gestore code può avere accesso esclusivo ai log e ai dati del gestore code condivisi. Quando ottiene l'accesso, diventa l'istanza attiva.

L'altra istanza che non riesce a ottenere l'accesso esclusivo attende come istanza in standby fino a quando i dati e i log del gestore code diventano disponibili.

Il file system di rete è responsabile del rilascio dei blocchi che detiene per l'istanza del gestore code attiva. Se l'istanza attiva ha esito negativo in qualche modo, il file system di rete rilascia i blocchi che sta detenendo per l'istanza attiva. Non appena il blocco esclusivo viene rilasciato, un gestore code in attesa del blocco tenta di acquisirlo. Se ha esito positivo, diventa l'istanza attiva e ha accesso esclusivo ai dati del gestore code e ai log sul file system condiviso. Continua quindi ad iniziare.

L'argomento correlato, [Supporto file system di pianificazione](#) descrive come impostare e verificare che il proprio file system supporti i gestori code a più istanze.

Un gestore code a più istanze non protegge l'utente da un errore nel filesystem. Esistono diversi modi per proteggere i tuoi dati.

- Investi in storage affidabile, come RAID (redundant disk array), e includili in un file system di rete con resilienza di rete.
- Eseguire il backup dei log lineari IBM MQ su un supporto alternativo e, se il supporto di log primario ha esito negativo, eseguire il ripristino utilizzando i log sul supporto alternativo. È possibile utilizzare un gestore code di backup per gestire questo processo.

Multi **Più istanze del gestore code**

Un gestore code a più istanze è resiliente perché utilizza un'istanza del gestore code in standby per ripristinare la disponibilità del gestore code dopo l'errore.

La replica delle istanze del gestore code è un modo molto efficiente per aumentare la disponibilità dei processi del gestore code. Utilizzando un modello di disponibilità semplice, puramente illustrativo: se l'affidabilità di un'istanza di un gestore code è del 99% (in un anno, il tempo di inattività cumulativo è di 3.65 giorni), l'aggiunta di un'altra istanza del gestore code aumenta la disponibilità a 99.99% (in un anno, il tempo di inattività cumulativo di circa un'ora).

Si tratta di un modello troppo semplice per fornire stime numeriche pratiche della disponibilità. Per modellare realisticamente la disponibilità, è necessario raccogliere statistiche per il tempo medio tra i guasti (MTBF) e il tempo medio di riparazione (MTTR), e la distribuzione di probabilità del tempo tra i guasti e i tempi di riparazione.

Il termine, gestore code a più istanze, fa riferimento alla combinazione di istanze attive e in standby del gestore code che condividono i dati e i log del gestore code. I gestori code a più istanze proteggono l'utente dall'errore dei processi del gestore code, in quanto un'istanza del gestore code è attiva su un server e un'altra istanza del gestore code è in standby su un altro server, pronta a subentrare automaticamente in caso di errore dell'istanza attiva.

Multi **Failover o commutazione**

Un'istanza del gestore code in standby subentra all'istanza attiva su richiesta (commutazione) o quando l'istanza attiva ha esito negativo (failover).

- Lo *switchover* si verifica quando un'istanza in standby viene avviata in seguito al comando **endmqm -s** emesso per l'istanza del gestore code attivo. È possibile specificare i **endmqm** parametri **-c**, **-i** o **-p** per controllare l'arresto improvviso del gestore code.

Nota: La commutazione ha luogo solo se un'istanza del gestore code in standby è già stata avviata. Il comando **endmqm -s** rilascia il blocco del gestore code attivo e consente la commutazione: non avvia un'istanza del gestore code in standby.

- Il *failover* si verifica quando il blocco dei dati del gestore code trattenuti dall'istanza attiva viene rilasciato perché l'istanza sembrava arrestarsi in modo imprevisto (ovvero, senza l'emissione di un comando **endmqm**).

Quando l'istanza in standby assume il controllo come istanza attiva, scrive un messaggio nel log degli errori del gestore code.

I client ricollegabili vengono riconnessi automaticamente quando un gestore code ha esito negativo o si commuta. Non è necessario includere l'indicatore **-x** nel comando **endmqm** per richiedere la

riconnessione client. La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

Se si rileva che non è possibile riavviare un'istanza non riuscita, anche se si è verificato un failover e l'istanza in standby è diventata attiva, verificare se le applicazioni connesse localmente all'istanza non riuscita si sono disconnesse dall'istanza non riuscita.

Le applicazioni connesse localmente devono terminare o disconnettersi da un'istanza del gestore code non riuscita per poter riavviare l'istanza non riuscita. Tutte le applicazioni connesse localmente che utilizzano i bind condivisi (che è l'impostazione predefinita) che mantengono una connessione a un'istanza non riuscita agiscono per impedire il riavvio dell'istanza.

Se non è possibile terminare le applicazioni connesse localmente o assicurarsi che si disconnettono quando l'istanza del gestore code locale ha esito negativo, considerare l'utilizzo di bind isolati. Le applicazioni connesse localmente che utilizzano collegamenti isolati non impediscono il riavvio dell'istanza del gestore code locale, anche se non si disconnettono.

Multi **Riconnessione canale e client**

La riconnessione del canale e del client è una parte essenziale del ripristino dell'elaborazione dei messaggi dopo che un'istanza del gestore code in standby è diventata attiva.

Le istanze del gestore code a più istanze sono installate su server con indirizzi di rete diversi. È necessario configurare i canali e client IBM MQ con le informazioni di connessione per tutte le istanze del gestore code. Quando subentra uno standby, i client e i canali vengono riconnessi automaticamente all'istanza del gestore code appena attiva al nuovo indirizzo di rete. La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

Il design è diverso da come funzionano gli ambienti ad alta disponibilità come HA - CMP. HA - CMP fornisce un indirizzo IP virtuale per il cluster e trasferisce l'indirizzo al server attivo. La riconnessione IBM MQ non modifica o reinstrada gli indirizzi IP. Funziona riconnettendosi utilizzando gli indirizzi di rete definiti nelle definizioni di canale e nelle connessioni client. In qualità di amministratore, è necessario definire gli indirizzi di rete nelle definizioni di canale e nelle connessioni client a tutte le istanze di qualsiasi gestore code a più istanze. Il modo migliore per configurare gli indirizzi di rete per un gestore code a più istanze dipende dalla connessione:

Canali gestore code

L'attributo CONNAME dei canali è un elenco separato da virgole di nomi di connessione; ad esempio CONNAME(' 127.0.0.1(1234) , 192.0.2.0(4321) '). Le connessioni vengono tentate nell'ordine specificato nell'elenco delle connessioni finché non viene stabilita una connessione con esito positivo. Se nessuna connessione ha esito positivo, il canale tenta di riconnettersi.

Canali cluster

Di norma, non è richiesta alcuna configurazione aggiuntiva per far funzionare i gestori code a più istanze in un cluster.

Se un gestore code si connette a un gestore code del repository, il repository rileva l'indirizzo di rete del gestore code. Fa riferimento al CONNAME del canale CLUSRCVR nel gestore code. Su TCP/IP, il gestore code imposta automaticamente il CONNAME se lo si omette o lo si configura su spazi vuoti. Quando un'istanza in standby prende il controllo, il relativo indirizzo IP sostituisce l'indirizzo IP della precedente istanza attiva come CONNAME.

Se necessario, è possibile configurare manualmente CONNAME con l'elenco di indirizzi di rete delle istanze del gestore code.

Connessioni client

Le connessioni client possono utilizzare elenchi di connessioni o gruppi di gestori code per selezionare connessioni alternative.

Quando si verifica il failover, la riconnessione richiede del tempo. Il gestore code in standby deve completare l'avvio. I client connessi al gestore code non riuscito devono rilevare l'errore di connessione e avviare una nuova connessione client. Se una nuova connessione client seleziona il gestore code in standby che è diventato nuovamente attivo, il client viene riconnesso allo stesso gestore code.

Se il client è nel mezzo di una chiamata MQI durante la riconnessione, deve tollerare un'attesa estesa prima del completamento della chiamata.

Se l'errore si verifica durante un trasferimento batch su un canale di messaggi, il batch viene sottoposto a rollback e riavviato.

La commutazione è più veloce del failover e impiega solo il tempo necessario per arrestare un'istanza del gestore code e avviarne un'altra. Per un gestore code con solo pochi record di log da ripetere, la commutazione potrebbe richiedere l'ordine di pochi secondi. Per stimare il tempo impiegato dal failover, è necessario aggiungere il tempo impiegato per rilevare l'errore. Nel migliore dei casi, il rilevamento richiede l'ordine di 10 secondi e potrebbe richiedere diversi minuti, a seconda della rete e del sistema di file.

Multi Ripristino applicazione

Il ripristino dell'applicazione è la continuazione automatizzata dell'elaborazione dell'applicazione dopo il failover. Il ripristino dell'applicazione dopo il failover richiede un'attenta progettazione. Alcune applicazioni devono essere a conoscenza del failover.

L'obiettivo del ripristino dell'applicazione è che l'applicazione continui l'elaborazione con un breve ritardo. Prima di continuare con la nuova elaborazione, l'applicazione deve eseguire il backout e inoltrare nuovamente l'unità di lavoro che stava elaborando durante l'errore.

Un problema per il recupero dell'applicazione è la perdita del contesto condiviso tra IBM MQ MQI client e il gestore code e memorizzato nel gestore code. IBM MQ MQI client ripristina la maggior parte del contesto, ma ci sono alcune parti del contesto che non possono essere ripristinate in modo affidabile. Le seguenti sezioni descrivono alcune proprietà del recupero dell'applicazione e come influiscono sul recupero delle applicazioni connesse a un gestore code a più istanze.

Messaggistica transazionale

Dalla prospettiva della consegna dei messaggi, il failover non modifica le proprietà persistenti della messaggistica IBM MQ. Se i messaggi sono persistenti e correttamente gestiti all'interno delle unità di lavoro, i messaggi non vengono persi durante un failover.

Dal punto di vista dell'elaborazione delle transazioni, le transazioni vengono sottoposte a backout o a commit dopo il failover.

Viene eseguito il rollback delle transazioni non sottoposte a commit. Dopo il failover, un'applicazione riconnettibile riceve un codice di errore MQRC_BACKED_OUT per indicare che la transazione non è riuscita. È quindi necessario riavviare nuovamente la transazione.

Le transazioni sottoposte a commit sono transazioni che hanno raggiunto la seconda fase di un commit a due fasi o transazioni a fase singola (solo messaggio) che hanno iniziato MQCMIT.

Se il gestore code è il coordinatore della transazione e MQCMIT ha iniziato la seconda fase del suo commit a due fasi prima dell'errore, la transazione viene completata correttamente. Il completamento è sotto controllo del gestore code e continua quando il gestore code è di nuovo in esecuzione. In un'applicazione ricollegabile, la chiamata MQCMIT viene completata normalmente.

In un commit a fase singola, che implica solo messaggi, una transazione che ha avviato l'elaborazione del commit viene completata normalmente sotto il controllo del gestore code una volta che è di nuovo in esecuzione. In un'applicazione ricollegabile, il MQCMIT viene completato normalmente.

I client ricollegabili possono utilizzare transazioni a fase singola sotto il controllo del gestore code come coordinatore delle transazioni. Il client transazionale esteso non supporta la riconnessione. Se la riconnessione viene richiesta quando il client transazionale si connette, la connessione ha esito positivo, ma senza la possibilità di riconnessione. La connessione si comporta come se non fosse ricollegabile.

Riavvio o ripresa dell'applicazione

Il failover interrompe un'applicazione. Dopo un errore, un'applicazione può essere riavviata dall'inizio oppure può riprendere l'elaborazione dopo l'interruzione. Quest'ultimo è chiamato *riconnessione client automatica*. La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

Con un'applicazione IBM MQ MQI client, è possibile impostare un'opzione di connessione per riconnettere automaticamente il client. Le opzioni sono MQCNO_RECONNECT o MQCNO_RECONNECT_Q_MGR. Se non è impostata alcuna opzione, il client non tenta di riconnettersi automaticamente e l'errore del gestore code restituisce MQRC_CONNECTION_BROKEN al client. È possibile progettare il client in modo che tenti di avviare una nuova connessione emettendo una nuova chiamata MQCONN o MQCONNX.

I programmi server devono essere riavviati; non possono essere riconnessi automaticamente dal gestore code nel momento in cui sono stati elaborati quando il gestore code o il server hanno avuto esito negativo. I programmi server IBM MQ in genere non vengono riavviati sull'istanza del gestore code in standby quando un'istanza del gestore code a più istanze ha esito negativo.

È possibile automatizzare il riavvio di un programma server IBM MQ sul server standby in due modi:

1. Impacchettare l'applicazione server come servizio gestore code. Viene riavviato al riavvio del gestore code in standby.
2. Scrivere la propria logica di failover, attivata ad esempio dal messaggio di log di failover scritto da un'istanza del gestore code in standby quando viene avviata. L'istanza dell'applicazione deve quindi richiamare MQCONN o MQCONNX dopo l'avvio, per creare una connessione al gestore code.

Rilevamento del failover

Alcune applicazioni devono essere a conoscenza del failover, altre no. Si considerino questi due esempi.

1. Un'applicazione di messaggistica che riceve o riceve messaggi su un canale di messaggistica normalmente non richiede che il gestore code all'altra estremità del canale sia in esecuzione: è improbabile che ne venga influenzata se il gestore code all'altra estremità del canale viene riavviato su un'istanza in standby.
2. Un'applicazione IBM MQ MQI client elabora l'input dei messaggi persistenti da una coda e inserisce le risposte dei messaggi persistenti in un'altra coda come parte di una singola unità di lavoro: se gestisce un codice di errore MQRC_BACKED_OUT da MQPUT, MQGET o MQCMIT all'interno del punto di sincronizzazione riavviando l'unità di lavoro, non viene perso alcun messaggio. Inoltre, l'applicazione non deve eseguire alcuna elaborazione speciale per gestire un errore di connessione.

Si supponga, tuttavia, nel secondo esempio, che l'applicazione stia esplorando la coda per selezionare il messaggio da elaborare utilizzando l'opzione MQGET, MQGMO_MSG_UNDER_CURSOR. La riconnessione reimposta il cursore di esplorazione e la chiamata MQGET non restituisce il messaggio corretto. In questo esempio, l'applicazione deve essere consapevole che si è verificato un failover. Inoltre, prima di emettere un altro MQGET per il messaggio sotto il cursore, l'applicazione deve ripristinare il cursore di ricerca.

La perdita del cursore di esplorazione è un esempio di come il contesto dell'applicazione cambia dopo la riconnessione. Altri casi sono documentati in [“Ripristino di un client riconnesso automaticamente”](#) a pagina 583.

Sono disponibili tre modelli di progettazione alternativi per le applicazioni IBM MQ MQI client in seguito a failover. Solo uno di essi non deve rilevare il failover.

Nessuna riconnessione

In questo modello, l'applicazione arresta tutta l'elaborazione sulla connessione corrente quando la connessione è interrotta. Per continuare l'elaborazione dell'applicazione, è necessario stabilire una nuova connessione con il gestore code. L'applicazione è interamente responsabile del trasferimento di tutte le informazioni di stato necessarie per continuare l'elaborazione sulla nuova connessione. Le applicazioni client esistenti che si riconnettono a un gestore code dopo aver perso la connessione vengono scritte in questo modo.

Il client riceve un codice motivo, ad esempio MQRC_CONNECTION_BROKEN o MQRC_Q_MGR_NOT_AVAILABLE dalla successiva chiamata MQI dopo la perdita della connessione. L'applicazione deve eliminare tutte le informazioni sullo stato IBM MQ, come gli handle di coda, ed emettere una nuova chiamata MQCONN o MQCONNX per stabilire una nuova connessione, quindi riaprire gli oggetti IBM MQ che deve elaborare.

Il comportamento MQI predefinito prevede che l'handle di connessione del gestore code diventi inutilizzabile dopo la perdita di una connessione con il gestore code. Il valore predefinito è equivalente all'impostazione dell'opzione MQCNO_RECONNECT_DISABLED su MQCONNX per impedire la riconnessione dell'applicazione dopo il failover.

Tolleranza failover

Scrivere l'applicazione in modo che non sia influenzata dal failover. A volte un'attenta gestione degli errori è sufficiente per gestire il failover.

Riconnessione consapevole

Registrare un gestore eventi MQCBT_EVENT_HANDLER con il gestore code. Il gestore eventi viene inviato con MQRC_RECONNECTING quando il client inizia a tentare di riconnettersi con il server e MQRC_RECONNECTED dopo una riconnessione eseguita correttamente. È quindi possibile eseguire una routine per ristabilire uno stato prevedibile in modo che l'applicazione del client possa continuare l'elaborazione.

Ripristino di un client riconnesso automaticamente

Il failover è un evento imprevisto e perché un client riconnesso automaticamente funzioni come progettato, le conseguenze della riconnessione devono essere prevedibili.

Un elemento importante per trasformare un errore imprevisto in un ripristino prevedibile e affidabile è l'utilizzo delle transazioni.

Nella sezione precedente, è stato fornito un esempio, “2” a pagina 582, di un IBM MQ MQI client che utilizza una transazione locale per coordinare MQGET e MQPUT. Il client emette una chiamata MQCMIT o MQBACK in risposta a un errore MQRC_BACKED_OUT e quindi inoltra nuovamente la transazione di cui è stato eseguito il backout. L'errore del gestore code causa il backout della transazione e il comportamento dell'applicazione client garantisce che non vengano perse transazioni e messaggi.

Si noti che per una richiamata, potrebbe essere necessario riprendere l'applicazione di consumo se lo stato del parametro del consumer di richiamata è: MQCS_SUSPENDED_USER_ACTION.

Non tutto lo stato del programma viene gestito come parte di una transazione e quindi le conseguenze della riconnessione diventano più difficili da comprendere. È necessario conoscere il modo in cui la riconnessione modifica lo stato di un IBM MQ MQI client per progettare l'applicazione client in modo che sopravviva al failover del gestore code.

Potresti decidere di progettare la tua applicazione senza alcun codice di failover speciale, gestendo gli errori di riconnessione con la stessa logica degli altri errori. In alternativa, è possibile scegliere di riconoscere che la riconnessione richiede un'elaborazione di errori speciali e registrare un gestore eventi con IBM MQ per eseguire una routine per gestire il failover. La routine potrebbe gestire l'elaborazione della riconnessione stessa oppure impostare un indicatore per indicare al thread del programma principale che quando riprende l'elaborazione è necessario eseguire l'elaborazione del recupero.

L'ambiente IBM MQ MQI client è consapevole del failover stesso e ripristina il maggior numero possibile di contesti, dopo la riconnessione, memorizzando alcune informazioni di stato nel client ed emettendo ulteriori chiamate MQI per conto dell'applicazione client per ripristinare il suo stato IBM MQ. Ad esempio, gli handle per gli oggetti che erano aperti nel punto di errore vengono ripristinati e le code dinamiche temporanee vengono aperte con lo stesso nome. Ma ci sono cambiamenti che sono inevitabili e hai bisogno del tuo design per affrontare questi cambiamenti. Le modifiche possono essere suddivise in cinque tipi:

1. Gli errori nuovi o precedentemente non diagnosticati vengono restituiti dalle chiamate MQI fino a quando non viene ripristinato un nuovo stato di contesto coerente dal programma di applicazione.

Un esempio di ricezione di un nuovo errore è il codice di ritorno MQRC_CONTEXT_NOT_AVAILABLE quando si tenta di passare il contesto dopo aver salvato il contesto prima della riconnessione. Il contesto non può essere ripristinato dopo la riconnessione perché il contesto di sicurezza non viene passato a un programma client non autorizzato. Per fare ciò, un programma di applicazione dannoso potrebbe ottenere il contesto di sicurezza.

In genere, le applicazioni gestiscono gli errori comuni e prevedibili in modo accurato e relegano gli errori non comuni in un gestore di errori generico. Il programma di gestione degli errori potrebbe disconnettersi da IBM MQ e riconnettersi di nuovo oppure arrestare del tutto il programma. Per migliorare la continuità, potrebbe essere necessario gestire alcuni errori in modo diverso.

2. I messaggi non persistenti potrebbero andare persi.
3. Viene eseguito il rollback delle transazioni (che possono anche sospendere i consumer asincroni, consultare il testo precedente).
4. Le chiamate MQGET o MQPUT utilizzate al di fuori di un punto di sincronizzazione potrebbero essere interrotte con la possibile perdita di un messaggio.
5. Errori di temporizzazione indotti, a causa di un'attesa prolungata in una chiamata MQI.

Alcuni dettagli sul contesto perso sono riportati nella seguente sezione.

- I messaggi non persistenti vengono eliminati, a meno che non vengano inseriti in una coda con l'opzione NPMCLASS (HIGH) e l'errore del gestore code non abbia interrotto l'opzione di memorizzazione dei messaggi non persistenti all'arresto.
 - Una sottoscrizione non durevole viene persa quando una connessione viene interrotta. Alla riconnessione, viene ristabilito. Considerare l'utilizzo di una sottoscrizione durevole.
 - L'intervallo get - wait viene ricalcolato; se il limite viene superato, restituisce MQRC_NO_MSG_AVAILABLE. Allo stesso modo, la scadenza della sottoscrizione viene ricalcolata per fornire la stessa scadenza globale.
 - La posizione del cursore di ricerca in una coda viene persa; in genere viene ristabilita prima del primo messaggio.
 - MQGET chiamate che specificano MQGMO_BROWSE_MSG_UNDER_CURSOR o MQGMO_MSG_UNDER_CURSOR, non riuscite con codice motivo MQRC_NO_MSG_AVAILABLE.
 - I messaggi bloccati per la ricerca sono sbloccati.
 - I messaggi contrassegnati con l'ambito dell'handle non sono contrassegnati e possono essere ricercati di nuovo.
 - Nella maggior parte dei casi, i messaggi contrassegnati da ricerca cooperativa non vengono contrassegnati.
 - Il contesto di sicurezza è andato perduto. I tentativi di utilizzare il contesto del messaggio salvato, come l'inserimento di un messaggio con MQPMO_PASS_ALL_CONTEXT, non riescono con MQRC_CONTEXT_NOT_AVAILABLE.
 - I token del messaggio vengono persi. MQGET utilizzando un token del messaggio restituisce il codice motivo MQRC_NO_MSG_AVAILABLE.
- Nota:** *MsgId* e *CorrelId*, poiché fanno parte del messaggio, vengono conservati con il messaggio durante il failover e quindi MQGET utilizzando *MsgId* o *CorrelId* funzionano come previsto.
- I messaggi inseriti su una coda nel punto di sincronizzazione in una transazione non sottoposta a commit non sono più disponibili.
 - L'elaborazione dei messaggi in un ordine logico o in un gruppo di messaggi, risulta in un codice di ritorno MQRC_RECONNECT_INCOMPATIBLE dopo la riconnessione.
 - Una chiamata MQI potrebbe restituire MQRC_RECONNECT_FAILED piuttosto che il più generale MQRC_CONNECTION_BROKEN che i client generalmente ricevono oggi.
 - La riconnessione durante una MQPUT chiamata all'esterno del punto di sincronizzazione restituisce MQRC_CALL_INTERRUPTED se IBM MQ MQI client non sa se il messaggio è stato consegnato correttamente al gestore code. La riconnessione durante MQCMIT funziona in modo simile.

- MQRC_CALL_INTERRUPTED viene restituito - dopo una riconnessione riuscita - se IBM MQ MQI client non ha ricevuto alcuna risposta dal gestore code per indicare l'esito positivo o negativo di
 - la consegna di un messaggio persistente utilizzando una chiamata MQPUT fuori dal punto di sincronizzazione.
 - la consegna di un messaggio persistente o di un messaggio con persistenza predefinita utilizzando una chiamata MQPUT1 fuori dal punto di sincronizzazione.
 - il commit di una transazione utilizzando una chiamata MQCMIT. La risposta viene restituita solo dopo una riconnessione riuscita.
- I canali vengono riavviati come nuove istanze (potrebbero anche essere canali differenti) e quindi non viene conservato alcuno stato di uscita del canale.
- Le code dinamiche temporanee vengono ripristinate come parte del processo di recupero dei client ricollegabili che avevano code dinamiche temporanee aperte. Non viene ripristinato alcun messaggio su una coda dinamica temporanea, ma le applicazioni che avevano la coda aperta o che avevano ricordato il nome della coda, sono in grado di continuare l'elaborazione.

Esiste la possibilità che se la coda viene utilizzata da un'applicazione diversa da quella che l'ha creata, potrebbe non essere ripristinata abbastanza rapidamente da essere presente al successivo riferimento. Ad esempio, se un client crea una coda dinamica temporanea come coda di risposta e un messaggio di risposta deve essere inserito sulla coda da un canale, la coda potrebbe non essere recuperata in tempo. In questo caso, il canale generalmente posiziona il messaggio di risposta sulla coda di messaggi non recapitabili.

Se un'applicazione client ricollegabile apre una coda dinamica temporanea in base al nome (perché un'altra applicazione l'ha già creata), quando si verifica la riconnessione, IBM MQ MQI client non è in grado di ricreare la coda dinamica temporanea perché non dispone del modello da cui crearla. In MQI, solo un'applicazione può aprire la coda dinamica temporanea per modello. Altre applicazioni che desiderano utilizzare la coda dinamica temporanea devono utilizzare MQPUT1 o i bind del server oppure essere in grado di provare nuovamente la riconnessione se non riesce.

Solo i messaggi non persistenti possono essere inseriti in una coda dinamica temporanea e questi messaggi vengono persi durante il failover; questa perdita si verifica per i messaggi inseriti in una coda dinamica temporanea utilizzando MQPUT1 durante la riconnessione. Se il failover si verifica durante MQPUT1, il messaggio potrebbe non essere inserito, anche se MQPUT1 ha esito positivo. Una soluzione temporanea a questo problema consiste nell'utilizzare code dinamiche permanenti. Qualsiasi applicazione di bind del server può aprire la coda dinamica temporanea in base al nome perché non è ricollegabile.

Multi **Data recovery e alta disponibilità**

Le soluzioni di alta disponibilità che utilizzano gestori code a più istanze devono includere un meccanismo per il ripristino dei dati dopo un errore di memoria.

Un gestore code a più istanze aumenta la disponibilità dei processi del gestore code, ma non la disponibilità di altri componenti, come ad esempio il file system, che il gestore code utilizza per memorizzare messaggi e altre informazioni.

Un modo per rendere i dati altamente disponibili è quello di utilizzare l'archiviazione dati resiliente in rete. È possibile creare una propria soluzione utilizzando un file system collegato in rete e un archivio dati resiliente oppure è possibile acquistare una soluzione integrata. Se si desidera combinare la resilienza con il ripristino di emergenza, è disponibile la replica del disco asincrona, che consente la replica del disco su decine o centinaia di chilometri.

È possibile configurare il modo in cui le diverse directory IBM MQ vengono associate ai supporti di archiviazione, per utilizzare al meglio i supporti. Per i gestori code *a più istanze*, esiste una distinzione importante tra due tipi di file e directory IBM MQ.

Directory che devono essere condivise tra le istanze di un gestore code.

Le informazioni che devono essere condivise tra diverse istanze di un gestore code si trovano in due directory: le directory `qmgrs` e `logs`. Le directory devono trovarsi su un file system di rete condiviso. Si consiglia di utilizzare un supporto di archiviazione che fornisce alta disponibilità continua

e prestazioni eccellenti perché i dati cambiano costantemente quando i messaggi vengono creati ed eliminati.

Le directory e i file che non hanno da condividere tra le istanze di un gestore code.

Alcune altre directory non devono essere condivise tra diverse istanze di un gestore code e vengono ripristinate rapidamente mediante mezzi diversi dall'utilizzo di un file system sottoposto a mirroring.

- File eseguibili IBM MQ e la directory tools. Sostituire reinstallando o eseguendo il backup e il ripristino da un archivio di file di cui è stato eseguito il backup.
- Informazioni di configurazione modificate per l'installazione nel suo insieme. Le informazioni di configurazione sono gestite da IBM MQ, come ad esempio il file `mqs.ini` sui sistemi AIX, Linux, and Windows, o fanno parte della propria gestione della configurazione, come ad esempio gli script di configurazione **MQSC**. Eseguire il backup e il ripristino utilizzando un archivio file.
- Output a livello di installazione come tracce, log degli errori e file FFDC. I file sono memorizzati nelle sottodirectory `errors` e `trace` nella directory di dati predefinita. La directory dei dati predefinita sui sistemi AIX and Linux è `/var/mqm`. Su Windows la directory dei dati predefinita è la directory di installazione IBM MQ.

È anche possibile utilizzare un gestore code di backup per eseguire backup di supporti regolari di un gestore code a più istanze utilizzando la registrazione lineare. Un gestore code di backup non fornisce un ripristino rapido come da un file system sottoposto a mirroring e non recupera le modifiche dall'ultimo backup. Il meccanismo del gestore code di backup è più appropriato per l'utilizzo in scenari di ripristino di emergenza offsite rispetto al ripristino di un gestore code dopo un malfunzionamento della memoria localizzata.

Combinazione di soluzioni IBM MQ Availability

Le applicazioni stanno utilizzando altre funzionalità IBM MQ per migliorare la disponibilità. I gestori code a più istanze completano altre capacità di alta disponibilità.

IBM MQ I cluster aumentano la disponibilità della coda

È possibile aumentare la disponibilità della coda creando più definizioni di una coda cluster; fino a un massimo di una coda su ciascun gestore nel cluster.

Si supponga che un membro del cluster abbia esito negativo e quindi venga inviato un nuovo messaggio a una coda cluster. A meno che il messaggio *non abbia* per passare al gestore code non riuscito, il messaggio viene inviato a un'altro gestore code in esecuzione nel cluster che ha una definizione della coda.

Anche se i cluster aumentano notevolmente la disponibilità, ci sono due scenari di errore correlati che provocano il ritardo dei messaggi. La creazione di un cluster con gestori code a più istanze riduce la probabilità che un messaggio venga ritardato.

Messaggi marooned

Se un gestore code nel cluster ha esito negativo, nessun altro messaggio che può essere instradato ad altri gestori code nel cluster viene instradato al gestore code non riuscito. I messaggi che sono già stati inviati vengono cancellati fino al riavvio del gestore code in errore.

Affinità

Affinità è il termine utilizzato per descrivere le informazioni condivise tra due calcoli altrimenti separati. Ad esempio, esiste un'affinità tra un'applicazione che invia un messaggio di richiesta a un server e la stessa applicazione che prevede di elaborare la risposta. Un altro esempio potrebbe essere una sequenza di messaggi, l'elaborazione di ogni messaggio in base ai messaggi precedenti.

Se si inviano messaggi alle code cluster, è necessario considerare le affinità. È necessario inviare messaggi successivi allo stesso gestore code oppure ogni messaggio può essere inviato a qualsiasi membro del cluster?

Se è necessario inviare messaggi allo stesso gestore code nel cluster e questo ha esito negativo, i messaggi attendono nella coda di trasmissione del mittente fino a quando il gestore code del cluster non è di nuovo in esecuzione.

Se il cluster è configurato con gestori code a più istanze, il ritardo nell'attesa del riavvio del gestore code non riuscito è limitato all'ordine di un minuto o più mentre lo standby prende il sopravvento. Quando lo standby è in esecuzione, i messaggi di cui è stato eseguito il marooned riprendono l'elaborazione, i canali per l'istanza del gestore code appena attivata vengono avviati e i messaggi che erano in attesa nelle code di trasmissione iniziano a fluire.

Un modo possibile per configurare un cluster per superare i messaggi ritardati da un gestore code non riuscito consiste nel distribuire due diversi gestori code a ciascun server nel cluster e fare in modo che uno sia attivo e uno sia l'istanza in standby dei diversi gestori code. Questa è una configurazione di standby attivo e aumenta la disponibilità del cluster.

Oltre a beneficiare di una gestione ridotta e di una maggiore scalabilità, i cluster continuano a fornire ulteriori elementi di disponibilità per integrare i gestori code a più istanze. I cluster proteggono da altri tipi di errori che influiscono sia sulle istanze attive che su quelle in standby di un gestore code.

Servizio ininterrotto

Un cluster fornisce un servizio ininterrotto. I nuovi messaggi ricevuti dal cluster vengono inviati ai gestori code attivi per l'elaborazione. Non fare affidamento su un gestore code a più istanze per fornire un servizio ininterrotto perché il gestore code in standby impiega del tempo per rilevare l'errore e completarne l'avvio, per riconnettere i canali e per inoltrare nuovamente i batch di messaggi non riusciti.

Interruzione localizzata


Esistono limitazioni pratiche alla distanza tra i server attivi, standby e file system, poiché devono interagire a velocità di millisecondi per fornire prestazioni accettabili.

I gestori code con cluster richiedono velocità di interazione dell'ordine di molti secondi e possono essere geograficamente distribuiti ovunque nel mondo.

Errore operativo

Utilizzando due meccanismi diversi per aumentare la disponibilità, si riducono le probabilità che un errore operativo, come un errore umano, comprometta i propri sforzi di disponibilità.

I gruppi di condivisione della coda aumentano la disponibilità di elaborazione dei messaggi

 I gruppi di condivisione code, forniti solo su z/OS, consentono a un gruppo di gestori code di condividere una coda. Se un gestore code ha esito negativo, gli altri gestori code continuano a elaborare tutti i messaggi sulla coda. I gestori code a più istanze non sono supportati su z/OS e integrano i gruppi di condivisione code solo come parte di un'architettura di messaggistica più ampia.

IBM MQ I client aumentano la disponibilità delle applicazioni

I programmi IBM MQ MQI client possono connettersi a gestori code differenti in un gruppo di gestori code in base alla disponibilità del gestore code, al peso della connessione e alle affinità. Eseguendo un'applicazione su una macchina differente rispetto a quella su cui è in esecuzione il gestore code, è possibile migliorare la disponibilità generale di una soluzione purché esista un modo per riconnettere l'applicazione se l'istanza del gestore code a cui è connessa non riesce.

I gruppi di gestori code vengono utilizzati per incrementare la disponibilità del client separando un client da un gestore code arrestato e bilanciando il carico delle connessioni client in un gruppo di gestori code, piuttosto che come un sprayer IP. L'applicazione client non deve avere alcuna affinità con il gestore code in errore, ad esempio una dipendenza da una particolare coda, oppure non può riprendere l'elaborazione.

La riconnessione automatica del client e i gestori code a più istanze aumentano la disponibilità del client risolvendo alcuni problemi di affinità. La riconnessione automatica del client non è supportata da IBM MQ classes for Java.

È possibile impostare l'opzione MQCNO MQCNO_RECONNECT_Q_MGRper forzare un client a riconnettersi allo stesso gestore code:

1. Se il gestore code a istanza singola precedentemente connesso non è in esecuzione, la connessione viene ritentata fino a quando il gestore code non è nuovamente in esecuzione.

2. Se il gestore code è configurato come gestore code a più istanze, il client si riconnette all'istanza attiva.

Ricollegandosi automaticamente allo stesso gestore code, vengono ripristinate molte delle informazioni di stato che il gestore code deteneva per conto del client, ad esempio le code che aveva aperto e l'argomento a cui aveva effettuato la sottoscrizione. Se il client ha aperto una coda di risposta dinamica per ricevere una risposta a una richiesta, viene ripristinata anche la connessione alla coda di risposta.

MQ Adv.

Linux

Alta disponibilità RDQM

RDQM (replicated data queue manager) è una soluzione alta disponibilità disponibile su piattaforme Red Hat Enterprise Linux per x86-64 .

Una configurazione RDQM è composta da tre server configurati in un gruppo HA (High Availability), ciascuno con una istanza del gestore code. Un'istanza è il gestore code in esecuzione, che replica in modo sincrono i dati nelle altre due istanze. Se il server su cui è in esecuzione questo gestore code ha esito negativo, viene avviata un'altra istanza del gestore code con cui devono operare i dati correnti. Le tre istanze del gestore code possono facoltativamente condividere un indirizzo IP mobile, quindi i client devono essere configurati solo con un singolo indirizzo IP. Solo un'istanza del gestore code può essere eseguita contemporaneamente, anche se il gruppo HA viene partizionato a causa di problemi di rete. Il server che esegue il gestore code è noto come 'primario', ciascuno degli altri due server è noto come 'secondario'.

Tre nodi sono utilizzati per ridurre notevolmente la possibilità che si verifichi una situazione di divisione del cervello. In un sistema ad alta disponibilità a due nodi, la divisione del cervello può verificarsi quando la connettività tra i due nodi è interrotta. Senza connettività, entrambi i nodi potrebbero eseguire il gestore code contemporaneamente, accumulando dati differenti. Quando la connessione viene ripristinata, ci sono due versioni differenti dei dati (un 'split-brain') e l'intervento manuale è richiesto per decidere quale serie di dati conservare e quale scartare.

RDQM utilizza un sistema a tre nodi con quorum per evitare la situazione di divisione del cervello. I nodi che possono comunicare con almeno uno degli altri nodi formano un quorum. I gestori code possono essere eseguiti solo su un nodo con quorum. Il gestore code non può essere eseguito su un nodo che non è connesso ad almeno un altro nodo, quindi non può essere eseguito su due nodi contemporaneamente:

- Se si verifica un malfunzionamento di un singolo nodo, il gestore code può essere eseguito su uno degli altri due nodi. Se due nodi hanno esito negativo, il gestore code non può essere eseguito sul nodo rimanente perché il nodo non ha il quorum (il nodo rimanente non è in grado di determinare se gli altri due nodi hanno avuto esito negativo o se sono ancora in esecuzione e hanno perso la connettività).
- Se un singolo nodo perde la connettività, il gestore code non può essere eseguito su questo nodo perché il nodo non ha il quorum. Il gestore code può essere eseguito su uno dei due nodi rimanenti, che hanno il quorum. Se tutti i nodi perdono la connettività, il gestore code non può essere eseguito su nessuno dei nodi, poiché nessuno dei nodi ha il quorum.

Nota: IBM MQ Console non supporta i gestori code di dati replicati. È possibile utilizzare IBM MQ Explorer con gestori code di dati replicati, ma questo non visualizza le informazioni specifiche delle funzioni RDQM.

La configurazione del gruppo dei tre nodi è gestita da Pacemaker. La replica tra i tre nodi è gestita da DRBD. (Consultare <https://clusterlabs.org/pacemaker/> per informazioni su Pacemaker e <https://docs.linbit.com/docs/users-guide-9.0/> per informazioni su DRBD.)

È possibile eseguire il backup dei gestori code di dati replicati utilizzando il processo descritto in “[Backup dei dati del gestore code](#)” a pagina 696. L'arresto del gestore code e il backup non ha alcun effetto sul monitoraggio del nodo eseguito dalla configurazione RDQM.

La seguente figura mostra una distribuzione tipica con un RDQM in esecuzione su ognuno dei tre nodi nel gruppo HA.

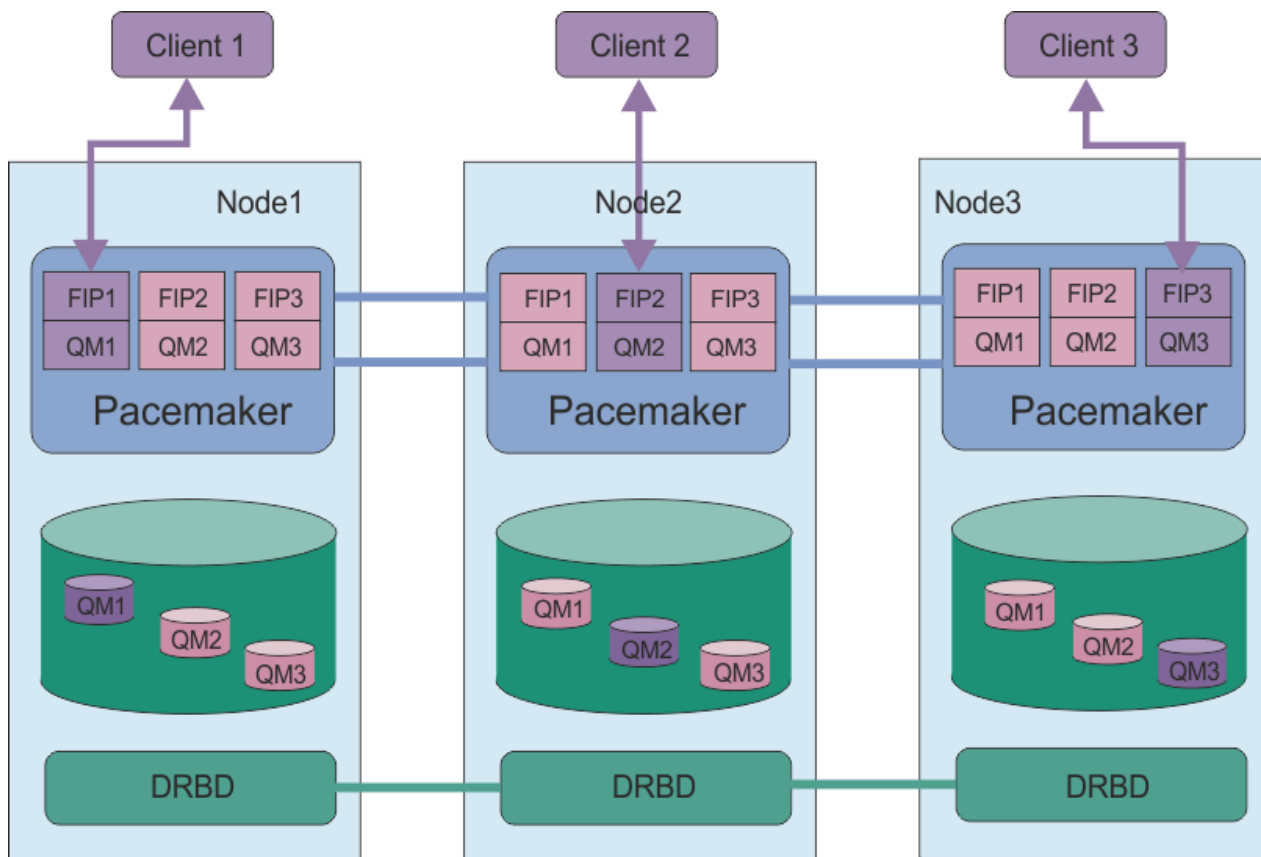


Figura 78. Esempio di gruppo HA con tre RDQM

Nella figura successiva, Node3 ha avuto esito negativo, i collegamenti Pacemaker sono stati persi e il gestore code QM3 viene eseguito su Node2 .

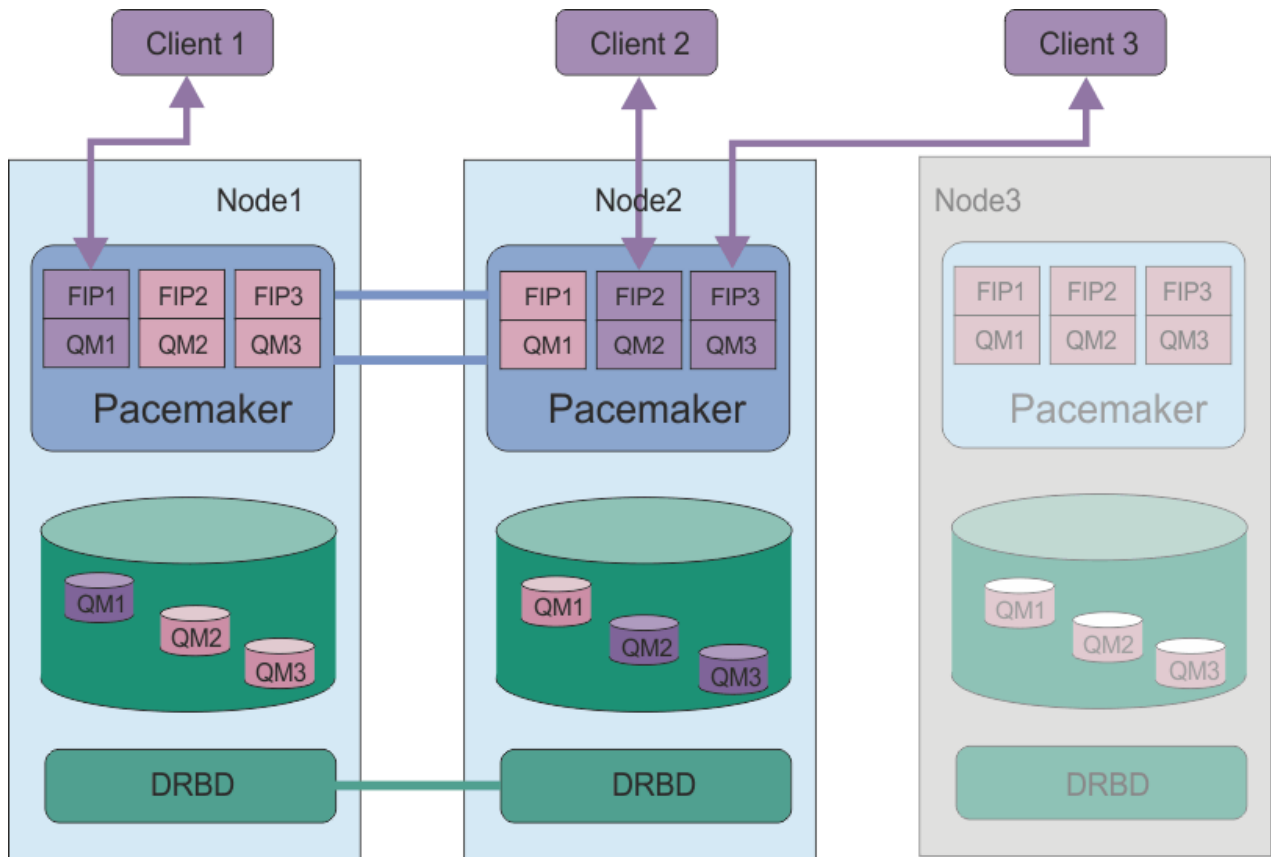


Figura 79. Esempio dopo l'esito negativo di node3

Nota: Quando i gestori code eseguono il failover su un altro nodo, conservano lo stato che avevano al failover. I gestori code in esecuzione vengono avviati, i gestori code arrestati rimangono arrestati.

Attività correlate

[Installazione di RDQM \(gestori code di dati replicati\)](#)

[Applicazione degli aggiornamenti del livello di manutenzione per RDQM](#)

[Migrazione dei gestori code di dati replicati](#)

[Risoluzione dei problemi relativi alle configurazioni RDQM](#)

Linux **Requisiti per la soluzione RDQM HA**

È necessario soddisfare diversi requisiti prima di configurare il gruppo HA (High Availability) RDQM.

Requisiti di sistema

Prima di configurare il gruppo HA RDQM, è necessario completare alcune operazioni di configurazione su ognuno dei tre server che fanno parte del gruppo HA.

- Ogni nodo richiede un gruppo di volumi denominato `drbdpool`. La memoria per ogni gestore code di dati replicati viene assegnata come un volume logico separato per gestore code da questo gruppo di volumi. Per prestazioni ottimali, questo gruppo di volumi deve essere costituito da uno o più volumi fisici che corrispondono alle unità disco interne (preferibilmente SSD). È possibile creare `drbdpool` prima o dopo aver installato la soluzione RDQM HA, ma è necessario creare `drbdpool` prima di creare effettivamente qualsiasi RDQM. Verificare la configurazione del gruppo di volumi utilizzando il comando `vgs`. L'output dovrebbe essere simile al seguente:

```
VG      #PV #LV #SN Attr   VSize  VFree
drbdpool 1   9  0 wz--n- <16.00g <7.00g
rhe1    1   2  0 wz--n- <15.00g  0
```

In particolare, verificare che non vi sia alcun carattere c nella sesta colonna degli attributi (ovvero, wz - - nc). c indica che il clustering è abilitato e, se lo è, è necessario eliminare il gruppo volumi e ricrearlo senza clustering.

- Dopo aver creato il gruppo di volumi `drbdpool`, non eseguire altre operazioni. IBM MQ gestisce i volumi logici creati in `drbdpool` come e dove sono montati.
- Ogni nodo richiede un massimo di tre interfacce utilizzate per configurare il supporto RDQM:
 - Un'interfaccia primaria per Pacemaker per monitorare il gruppo HA.
 - Un'interfaccia alternativa per Pacemaker per monitorare il gruppo HA.
 - Un'interfaccia per la replica di dati sincrona, nota come interfaccia di replica. Deve avere una larghezza di banda sufficiente per supportare i requisiti di replica dato il carico di lavoro previsto di tutti i gestori code di dati replicati in esecuzione nel gruppo HA.

È possibile configurare il gruppo HA in modo che lo stesso indirizzo IP venga utilizzato per tutte e tre le interfacce, un indirizzo IP separato viene utilizzato per ciascuna interfaccia o lo stesso indirizzo IP viene utilizzato per l'indirizzo principale e alternativo e un indirizzo IP separato per l'interfaccia di replica.

Per la massima tolleranza di errore, queste interfacce devono essere NIC (Network Interface Card) indipendenti.

- DRBD richiede che ogni nodo nel gruppo HA abbia un nome host internet valido (il valore restituito da `uname -n`), come definito da RFC 952 modificato da RFC 1123.
- Se è presente un firewall tra i nodi nel gruppo HA, il firewall deve consentire il traffico tra i nodi su un intervallo di porte. Viene fornito uno script di esempio, `/opt/mqm/samp/rdqm/firewalld/configure.sh`, che apre le porte necessarie se si sta eseguendo il firewall standard in RHEL. È necessario eseguire lo script come `root`. Se si utilizza un altro firewall, esaminare le definizioni di servizio `/usr/lib/firewalld/services/rdqm*` per vedere quali porte devono essere aperte. Lo script aggiunge le regole di servizio firewallD permanenti per DRBD, Pacemaker e IBM MQ:
 - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml` consente le porte TCP 7000-7100.
 - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-pacemaker.xml` consente le porte UDP 5404-5407
 - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml` consente la porta TCP 1414 (è necessario modificare lo script se si richiede una porta diversa)
- Se il sistema utilizza SELinux in modalità di forzatura, potrebbe essere necessario eseguire il seguente comando:

```
semanage permissive -a drbd_t
```

Se è stato installato il pacchetto `drbd-selinux`, non è necessario eseguire **semanage**. È necessario che questo pacchetto sia installato su ogni nodo oppure eseguire **semanage** su ogni nodo.

Requisiti di rete

Si consiglia di individuare i tre nodi nel gruppo HA RDQM nello stesso centro dati.

Se si sceglie di individuare i nodi in data center differenti, tenere presente le seguenti limitazioni:

- Le prestazioni peggiorano rapidamente con l'aumento della latenza tra i data center. Anche se IBM supporterà una latenza fino a 5 ms, potresti scoprire che le prestazioni della tua applicazione non possono tollerare più di 1 o 2 ms di latenza.
- I dati inviati attraverso il link di replicazione non sono soggetti ad alcuna ulteriore crittografia oltre a quella che potrebbe essere in atto utilizzando IBM MQ AMS.

Facoltativamente, è possibile configurare un indirizzo IP mobile per consentire a un client di utilizzare lo stesso indirizzo IP per un gestore code di dati replicati (RDQM) indipendentemente dal nodo nel gruppo HA su cui è in esecuzione. L'indirizzo mobile si collega a un'interfaccia fisica denominata sul nodo primario per RDQM. Se l'RDQM esegue il failover e un nodo differente diventa il nodo primario, l'IP mobile viene

collegato a un'interfaccia con lo stesso nome sul nuovo primario. Le interfacce fisiche sui tre nodi devono tutti avere lo stesso nome e appartenere alla stessa sottorete dell'indirizzo IP mobile.

Requisiti utente per configurare il cluster

È possibile configurare il gruppo HA RDQM come utente `root`. Se non si desidera eseguire la configurazione come `root`, eseguire la configurazione come utente nel gruppo `mqm`. Per un utente del gruppo `mqm` per configurare il cluster RDQM, è necessario soddisfare i requisiti riportati di seguito:

- L'utente `mqm` deve essere in grado di utilizzare `sudo` per eseguire comandi su ciascuno dei tre server che costituiscono il gruppo HA RDQM.
- Se l'utente `mqm` può utilizzare SSH senza una password per eseguire i comandi su ciascuno dei tre server che costituiscono il gruppo HA RDQM, l'utente deve eseguire i comandi solo su uno dei server.
- L'utente `mqm` deve avere lo stesso UID su tutti e tre i server.
- Il gruppo `mqm` deve avere lo stesso GID su tutti i tre server.

È necessario configurare `sudo` in modo che l'utente `mqm` possa eseguire questi comandi con autorità `root`:

```
/opt/mqm/bin/crtmqm  
/opt/mqm/bin/dltmqm  
/opt/mqm/bin/rdqmadm  
/opt/mqm/bin/rdqmstatus
```

Requisiti utente per l'utilizzo dei gestori code

Per creare, eliminare o configurare i gestori code di dati replicati (RDQM), è necessario utilizzare un ID utente che appartiene ai gruppi `mqm` e `haclient` (il gruppo `haclient` viene creato durante l'installazione di Pacemaker).

Linux

Impostazione dell'accesso SSH e sudo senza password

Puoi configurare SSH senza password e l'accesso `sudo` in modo da dover immettere solo i comandi di configurazione su un nodo nel gruppo HA. (L'impostazione di tale accesso è facoltativa, in alternativa è possibile eseguire i comandi su ciascun nodo.)

Informazioni su questa attività

Per impostare SSH senza password, devi configurare l'id `mqm` su ogni nodo, quindi generare una chiave su ogni nodo per quell'utente. Si distribuiscono quindi le chiavi agli altri nodi e si verifica la connessione per aggiungere ciascun nodo all'elenco di host noti. Infine, si blocca l'id `mqm`.

Nota: Le istruzioni presuppongono che si stia definendo un gruppo HA con interfacce primarie, alternative e di replica separate e che si definisca quindi l'accesso SSH senza password sulle interfacce primaria e alternativa. Se intendi configurare un sistema con un singolo indirizzo IP, definisci l'accesso SSH senza password su quella singola interfaccia. Se si prevede di configurare un sistema con due indirizzi IP per `HA_Primary` e `HA_Replication`, l'`ssh` deve essere configurato per l'indirizzo `HA_Primary`.

Puoi quindi creare l'accesso `sudo` per l'id `mqm` su ogni nodo.

Procedura

1. Per impostare SSH senza password:
 - a) Su ciascuno dei tre nodi, completa la seguente procedura per configurare l'utente `mqm` e generare una chiave SSH:
 - i) Modificare la directory home di `mqm` in `/home/mqm`:

```
use::mod -d /home/mqm mqm
```


ii) Creare la directory /home/mqm :

```
mkhomedir_helper mqm
```

iii) Aggiungere la password mqm :

```
passwd mqm
```

iv) Eseguire la shell interattiva come mqm:

```
su mqm
```

v) Generare la chiave di autenticazione mqm :

```
ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
```

b) Su ciascuno dei tre nodi, completare la seguente procedura per aggiungere la chiave del nodo agli altri due nodi e verificare le connessioni per ogni nodo primario e (se utilizzato) gli indirizzi alternativi:

i) Aggiungere la chiave ai nodi remoti:

```
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
```

ii) Controllare ssh senza password e aggiornare known_hosts per i nodi remoti:

```
ssh remote_node1_primary_address uname -n
ssh remote_node1_alternate_address uname -n
ssh remote_node2_primary_address uname -n
ssh remote_node2_alternate_address uname -n
```

Per ogni connessione, viene richiesto di confermare che si desidera procedere. Confermare per ognuno di aggiornare known_hosts. È necessario completare questa operazione prima di tentare di configurare il gruppo HA utilizzando SSH senza password.

iii) Uscire dalla shell interattiva come mqm:

```
exit
```

c) Su ciascun nodo, come root, completare la seguente procedura per eliminare la password mqm e bloccare l'ID:

i) Rimuovere la password mqm :

```
passwd -d mqm
```

ii) Blocca mqm:

```
passwd -l mqm
```

2. Su ogni nodo, come root, impostare l'accesso sudo per l'utente mqm creando il file /etc/sudoers.d/mqm contenente il seguente testo:

```
mqm ALL=(root) NOPASSWD: /opt/mqm/bin/crtmqm, /opt/mqm/bin/dltmqm, /opt/mqm/bin/rdqmadm, /opt/mqm/bin/rdqmstatus, /opt/mqm/bin/rdqmdr
```

Linux **Definizione del cluster Pacemaker (gruppo HA)**

Il gruppo HA è un cluster Pacemaker . Definire il cluster Pacemaker modificando il file /var/mqm/rdqm.ini ed eseguendo il comando **rdqmadm** .

Informazioni su questa attività

Consultare <https://clusterlabs.org/pacemaker/> per informazioni su Pacemaker. È possibile creare il cluster Pacemaker come utente nel gruppo `mqm` se l'utente `mqm` può utilizzare `sudo`. Se l'utente può anche eseguire l'SSH su ciascun server senza una password, è necessario modificare solo il file `rdqm.ini` ed eseguire `rdqmadm` su uno dei server per creare il cluster Pacemaker. Altrimenti, è necessario creare il file ed eseguire il comando come `root` su ciascuno dei server che devono essere nodi.

Il file `rdqm.ini` fornisce gli indirizzi IP utilizzati da RDQM per i nodi nel cluster Pacemaker. È necessario fornire il nome di ciascun nodo, che deve corrispondere al nome host come restituito dal comando `uname -n`.

Un gruppo HA RDQM può essere configurato per utilizzare uno, due o tre indirizzi IP:

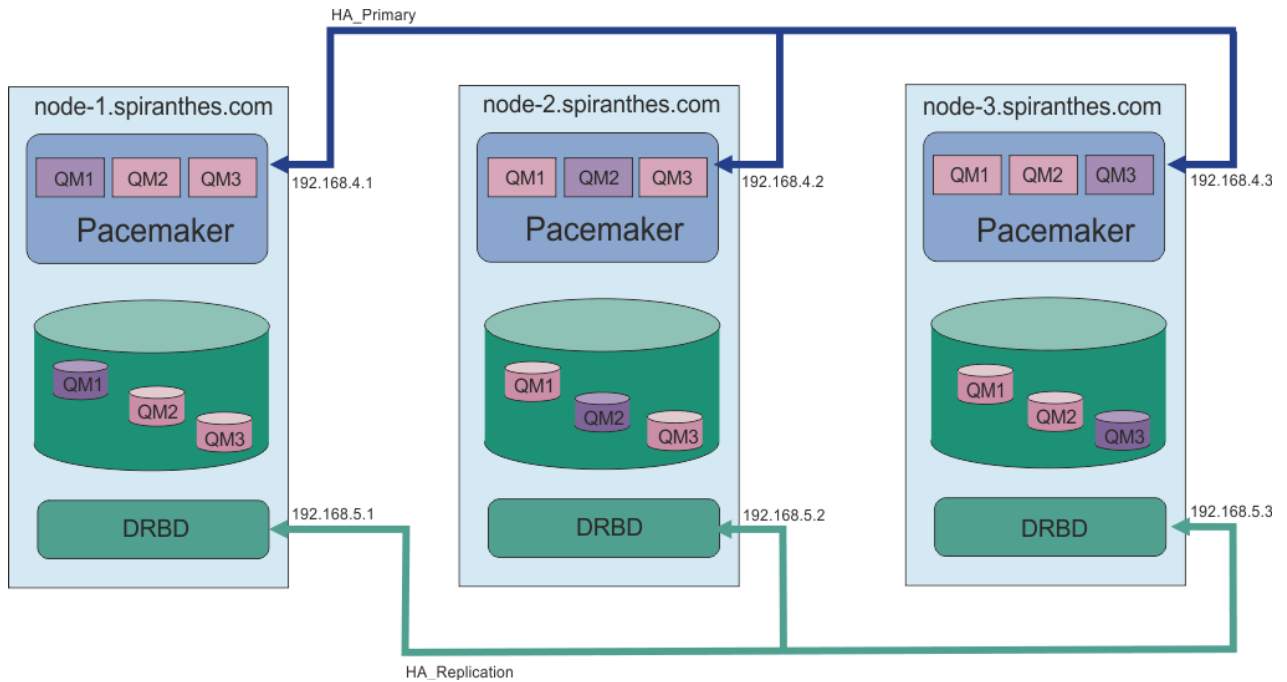
- Un indirizzo IP: Heartbeats e replica condividono lo stesso link
- Due indirizzi IP: i heartbeat e la replica utilizzano collegamenti separati
- Tre indirizzi IP: un link per la replica e due link separati per gli heartbeat

Queste opzioni vengono fornite per supportare diversi pattern di distribuzione per RDQM. Le diverse opzioni possono essere utilizzate per massimizzare la resilienza della soluzione RDQM in base all'ambiente utilizzato. Le configurazioni che utilizzano due o tre indirizzi IP sono principalmente destinate per le distribuzioni in cui è richiesto il controllo granulare su quali reti fisiche collegano gli heartbeat e il traffico di replica utilizzano per configurare la ridondanza per la connettività tra i nodi. In alternativa, la connettività altamente disponibile e resiliente può essere implementata a livello di rete, ad esempio utilizzando l'aggregazione di link. Con l'aggregazione di link, vengono utilizzati più link di rete fisici per fornire un singolo link logico che può continuare a funzionare se i singoli link fisici non riescono. Se RDQM viene distribuito in un ambiente in cui la connettività di rete è virtualizzata e / o in cui la connettività resiliente è implementata a livello di rete, è preferibile utilizzare un singolo indirizzo IP sia per gli heartbeat che per la replica.

Il seguente esempio illustra l'utilizzo di due indirizzi IP. Il file `rdqm.ini` ha un campo `HA_Primary` e un campo `HA_Replication` per ogni nodo, ma nessun campo `HA_Alternate`:

```
Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

Il seguente diagramma illustra questa configurazione:



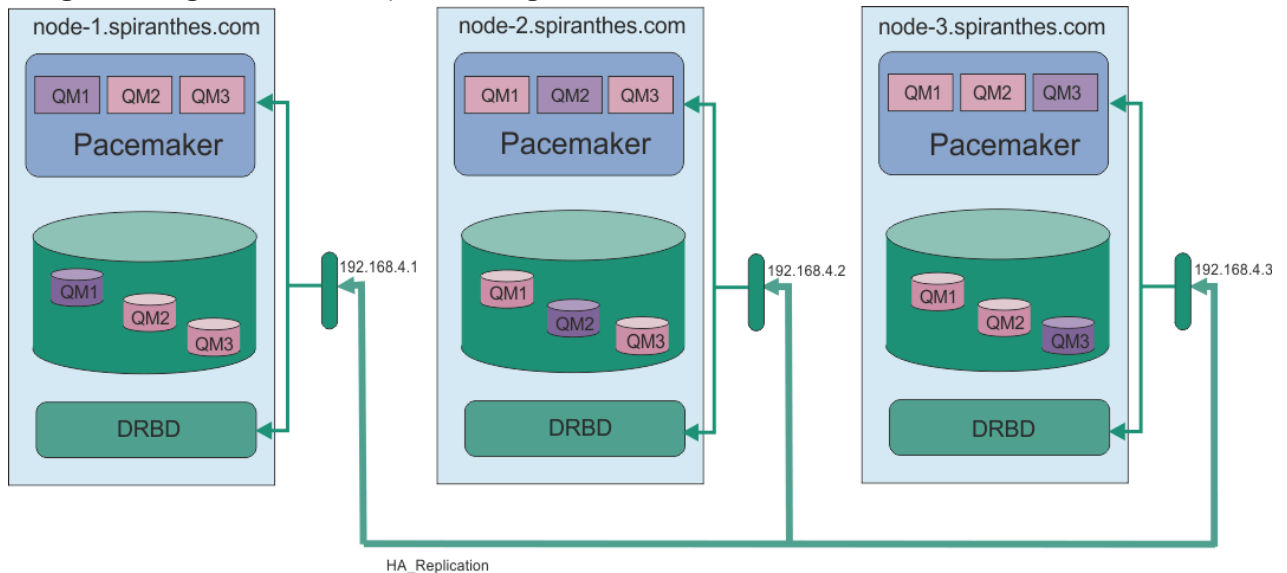
Il seguente file di esempio mostra la configurazione per un esempio di cluster Pacemaker che usa l'interfaccia HA_Replication per il monitoraggio. In questo caso, specificare solo l'interfaccia HA_Replication :

```

Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Replication=192.168.4.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Replication=192.168.4.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Replication=192.168.4.3

```

Il seguente diagramma illustra questa configurazione:



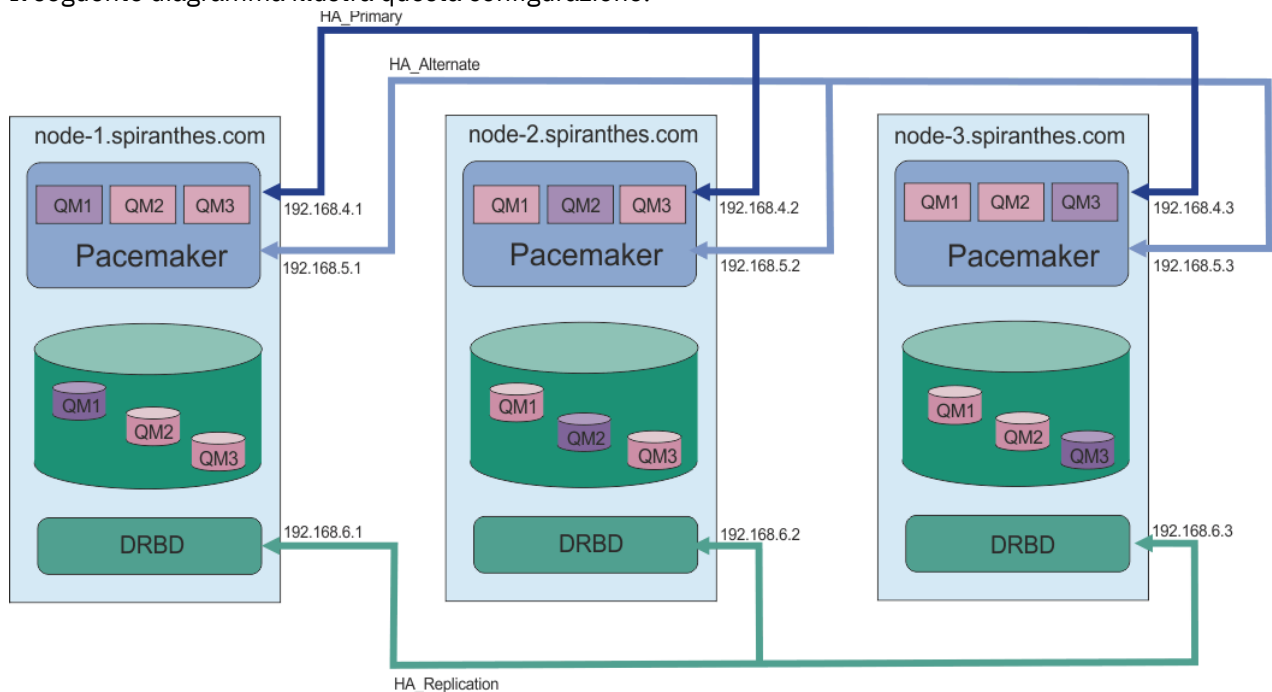
Il seguente file di esempio mostra la configurazione per un cluster Pacemaker di esempio che utilizza un indirizzo IP separato per ogni interfaccia:

```

Node:
  Name=rdqm-node-1.spiranthes.com
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  Name=rdqm-node-2.spiranthes.com
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  Name=rdqm-node-3.spiranthes.com
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3

```

Il seguente diagramma illustra questa configurazione:



L'ordine in cui si specificano i nodi deve essere lo stesso in tutti i file `rdqm.ini` nella configurazione. I tre nodi devono avere una vista comune per cui uno è Node1, un Node2 e così via.

Procedura

- Per definire il cluster Pacemaker come utente `root`:
 - a) Modificare il file `/var/mqm/rdqm.ini` su uno dei tre server in modo che il file definisca il cluster.
 - b) Copiare il file negli altri due server che saranno nodi nel cluster Pacemaker .
 - c) Eseguire il seguente comando come `root` su ognuno dei tre server:

```
rdqmadm -c
```

- Per definire il cluster Pacemaker come utente nel gruppo `mqm` su ciascun nodo:
 - a) Assicurarsi che l'utente `mqm` possa utilizzare **sudo** per eseguire comandi.
 - b) Modificare il file `/var/mqm/rdqm.ini` su uno dei tre server in modo che il file definisca il cluster Pacemaker .
 - c) Copiare `/var/mqm/rdqm.ini` negli altri due server che saranno nodi nel cluster Pacemaker .
 - d) Immettere il seguente comando su ciascun server:

```
rdqmadm -c
```

- Per definire il cluster Pacemaker come utente nel gruppo mqm da un nodo:
 - a) Assicurarsi che l'utente mqm possa utilizzare **sudo** per eseguire i comandi e possa connettersi a ogni server utilizzando SSH senza una password.
 - b) Modificare il file `/var/mqm/rdqm.ini` su uno dei tre server in modo che il file definisca il cluster Pacemaker .
 - c) Esegui il seguente comando:

```
rdqmadm -c
```

Riferimenti correlati

[rdqmadm \(gestione cluster gestore code dati replicati\)](#)

Linux *Eliminazione del cluster Pacemaker (gruppo HA)*

Il gruppo HA è un cluster Pacemaker . È possibile eliminare una configurazione cluster Pacemaker eseguendo il comando **rdqmadm** con l'opzione `-u` .

Informazioni su questa attività

Non è possibile eliminare la configurazione del cluster Pacemaker se i gestori code di dati replicati esistono ancora su uno dei nodi.

Procedura

- Per eliminare la configurazione del cluster Pacemaker , immettere il seguente comando da uno dei nodi:

```
rdqmadm -u
```

Riferimenti correlati

[rdqmadm \(gestione cluster gestore code dati replicati\)](#)

Linux *Creazione di un RDQM HA*

Utilizzare il comando **crtmqm** per creare un gestore code di dati replicati ad alta disponibilità (RDQM).

Informazioni su questa attività

È possibile creare un gestore code dati replicati ad alta disponibilità (RDQM) come utente nel gruppo mqm se l'utente mqm può utilizzare sudo. Se l'utente può anche eseguire SSH su ogni nodo senza una password, è necessario eseguire il comando di creazione RDQM solo su un nodo per creare RDQM su tutti e tre i nodi. Altrimenti, è necessario essere `root` per creare un RDQM ed eseguire i comandi su tutti e tre i nodi.

Nota: Esiste un limite assoluto di 129 gestori code in un gruppo HA. Se si tenta di creare più di questo, il tentativo avrà esito negativo. In pratica, l'aggiunta di più di 50 gestori code a un gruppo HA potrebbe riscontrare problemi di timeout.

I seguenti punti forniscono alcune indicazioni sul dimensionamento del file system del gestore code:

1. Durante la creazione di un gestore code RDQM, viene allocato un file system per memorizzare i dati e i log del gestore code. È importante dimensionare questo file system in modo appropriato in modo che il gestore code possa registrare l'attività in corso nei propri log e memorizzare i messaggi dell'applicazione nelle code. Quando si ridimensiona il file system, considerare i requisiti di messaggistica di picco, la crescita futura del carico di lavoro e le interruzioni dell'applicazione che potrebbero causare l'accumulo di messaggi nelle code. Per istruzioni sul calcolo della dimensione del log di ripristino del gestore code, consultare [“Quanto deve essere grande il file system del log?” a pagina 675](#). Quando si calcolano i requisiti di memoria per i messaggi dell'applicazione, è necessario

considerare la dimensione e il numero di messaggi, l'intestazione MQMD e le proprietà del messaggio di cui dispongono.

2. I file system del gestore code RDQM non possono essere ridimensionati dinamicamente. È necessario eseguire il backup e ripristinare un gestore code RDQM con un file system più grande, se richiesto, consultare [“Ridimensionamento del file system per un gestore code HA RDQM”](#) a pagina 603.
3. È possibile limitare la dimensione delle singole code sul disco utilizzando attributi della coda locale, come MAXDEPTH e MAXFSIZE. Consultare [Modifica dei file della coda IBM MQ](#).
4. È necessario monitorare l'utilizzo del disco in corso e rispondere in modo appropriato se l'utilizzo del disco aumenta prima che l'utilizzo del file system diventi critico. L'utilizzo del file system può essere monitorato utilizzando le funzionalità della piattaforma / del sistema operativo o sottoscrivendo le metriche pubblicate negli argomenti di sistema IBM MQ descritti in [Metriche pubblicate negli argomenti di sistema](#).

Procedura

- Per creare un RDQM come utente nel gruppo mqm :
 - a) Assicurarsi che l'utente mqm possa utilizzare **sudo** per eseguire i comandi e che possa connettersi a ciascun server utilizzando SSH senza una password.
 - b) Immettere il seguente comando:

```
crtmqm -sx [-fs FilesystemSize] qmname
```

dove *qmname* è il nome del gestore code di dati replicati. Facoltativamente, è possibile specificare la dimensione del file system per il gestore code (ossia, la dimensione del volume logico creato nel gruppo di volumi drbdpool).

Il comando tenta di utilizzare SSH per connettersi agli altri nodi nel cluster come utente mqm . Se la connessione ha esito positivo, le istanze secondarie del gestore code vengono create sui nodi. In caso contrario, è necessario creare le istanze secondarie ed eseguire il comando **crtmqm -sx** (come descritto per l'utente root).

- Per creare un RDQM come utente root:
 - a) Immettere il comando riportato di seguito su ciascuno dei nodi che devono ospitare le istanze secondarie di RDQM:

```
crtmqm -sxs [-fs FilesystemSize] qmname
```

dove *qmname* è il nome del gestore code di dati replicati. Facoltativamente, è possibile specificare la dimensione del file system per il gestore code (ossia, la dimensione del volume logico creato nel gruppo di volumi drbdpool). È necessario specificare la stessa dimensione del file system per RDQM su tutti e tre i nodi nel gruppo HA. La dimensione è un valore numerico, specificato in GB. È possibile specificare un valore in MB immettendo il valore seguito dal carattere M

Il comando crea un'istanza secondaria di RDQM.

- b) Sul nodo rimanente, immettere il seguente comando:

```
crtmqm -sx [-fs FilesystemSize] qmname
```

dove *qmname* è il nome del gestore code di dati replicati. Facoltativamente, è possibile specificare la dimensione del filesystem per il gestore code. La dimensione è un valore numerico, specificato in GB. È possibile specificare un valore in MB immettendo il valore seguito dal carattere M.

Il comando determina se l'istanza secondaria del gestore code esiste sugli altri due nodi. Se esistono comandi secondari, il comando crea e avvia il gestore code primario. Se i secondari non esistono, viene richiesto di eseguire il comando **crtmqm -sxs** su ciascuno dei nodi.

Oltre agli argomenti DataPath (**-md**) e LogPath (**-ld**), tutti gli argomenti validi per la creazione di un gestore code standard Linux sono validi anche per un gestore code di dati replicati primario.

Nota: Quando si crea un RDQM, per il link di replica viene assegnato il successivo numero di porta libero superiore a 7000. Se viene rilevato che la porta scelta è utilizzata da un'altra applicazione, il comando **crtmqm** ha esito negativo con l'errore AMQ6543 e tale porta viene aggiunta a un elenco di esclusione. È possibile eliminare le istanze secondarie del gestore code, quindi eseguire nuovamente il comando **crtmqm**.

Riferimenti correlati

[crtmqm](#)

Linux *Eliminazione di un RDQM HA*

Utilizzare il comando **dltmqm** per eliminare un RDQM (high availability replicated data queue manager).

Informazioni su questa attività

È necessario eseguire il comando per eliminare RDQM sul nodo primario di RDQM. RDQM deve essere terminato per primo. È possibile eseguire il comando come un utente mqm se tale utente dispone dei privilegi sudo necessari. Altrimenti, è necessario eseguire il comando come root. Una volta eliminate le risorse associate al gestore code primario, il comando tenta di eliminare i gestori code secondari utilizzando ssh per connettersi agli altri nodi. Se questa eliminazione non riesce, è necessario eseguire **dltmqm** manualmente sugli altri nodi per completare il processo. Su un nodo secondario, il comando ha esito negativo se il gestore code primario non è stato già eliminato.

Procedura

- Per eliminare un RDQM, immettere il seguente comando:

```
dltmqm RDQM_name
```

Riferimenti correlati

[dltmqm](#)

MQ Adv. **Linux** *Migrazione di un gestore code per diventare un gestore code HA RDQM*

È possibile migrare un gestore code esistente per diventare un gestore code di dati replicati (RDQM) HA (high availability) eseguendo il backup dei relativi dati persistenti, quindi ripristinando i dati in un gestore code RDQM appena creato con lo stesso nome.

Informazioni su questa attività

I gestori code di dati replicati HA richiedono un volume logico dedicato (file system) e la configurazione della replica del disco e del controllo HA. Questi componenti vengono configurati solo quando viene creato un nuovo gestore code. Un gestore code esistente può essere migrato per utilizzare RDQM eseguendo il backup dei dati persistenti, quindi ripristinando i dati su un gestore code RDQM appena creato con lo stesso nome. Questa procedura conserva la configurazione del gestore code, lo stato e i messaggi persistenti al momento della creazione del backup.

Nota: È possibile migrare un gestore code solo da una versione di IBM MQ uguale o inferiore alla versione in cui è installato RDQM. Anche il sistema operativo e l'architettura devono essere uguali. Altrimenti, è necessario creare un nuovo gestore code sulla piattaforma di destinazione; fare riferimento a [Spostamento di un gestore code su un sistema operativo differente](#).

È necessario soddisfare le seguenti condizioni prima di migrare un gestore code:

- Valutare i requisiti di alta disponibilità e consultare [“Alta disponibilità RDQM”](#) a pagina 588.
- Esaminare le applicazioni e i gestori code che si connettono al gestore code. Considerare le modifiche richieste per instradare le connessioni al nodo RDQM su cui è in esecuzione il gestore code. Ad esempio, se si configura l'alta disponibilità RDQM, è possibile considerare l'utilizzo di un indirizzo IP mobile, consultare [“Creazione ed eliminazione di un indirizzo IP mobile”](#) a pagina 606.
- Eseguire il provisioning o identificare i nodi RDQM esistenti per la configurazione scelta. Per informazioni relative ai requisiti di sistema per RDQM, consultare [“Requisiti per la soluzione RDQM HA”](#) a pagina 590.

- Installare IBM MQ Advanced, che include la funzione RDQM, su ciascun nodo.
- Configurare la configurazione del gruppo HA RDQM, consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593.
- Facoltativamente, verificare la configurazione RDQM utilizzando un gestore code di prova, che può quindi essere eliminato. La verifica della configurazione è consigliata per identificare e risolvere eventuali problemi prima di migrare il gestore code.
- Esaminare la configurazione della sicurezza per il gestore code, quindi replicare gli utenti e i gruppi locali richiesti su ciascun nodo RDQM.
- Esaminare il gestore code e la configurazione del canale per determinare se vengono utilizzate le uscite API, le uscite canale o le uscite di conversione dati. Installare le uscite richieste su ciascun nodo RDQM.
- Esaminare i servizi del gestore code che sono stati definiti, quindi installare e configurare i processi richiesti su ciascun nodo RDQM.

Procedura

1. Eseguire il backup del gestore code esistente:

- a) Arrestare il gestore code esistente emettendo un comando di attesa arresto `endmqm -wo` un comando di arresto immediato `endmqm -i`. Questa operazione è importante per garantire la congruenza dei dati nel backup.
- b) Determinare l'ubicazione della directory dei dati del gestore code visualizzando il file di configurazione IBM MQ, `mqm.ini`. Su Linux, questo file si trova nella directory `/var/mqm`. Per ulteriori informazioni su `mqm.ini`, consultare [“File di configurazione IBM MQ, mqm.ini”](#) a pagina 94.

Individuare la sezione `QueueManager` per il gestore code nel file. Se la stanza contiene una chiave denominata `DataPath`, il suo valore è la directory dei dati del gestore code. Se la chiave non esiste, la directory dei dati del gestore code può essere determinata utilizzando i valori delle chiavi `Prefix` e `Directory`. La directory dei dati del gestore code è una concatenazione di questi valori, nel formato `prefisso/qmgrs/directory`. Per ulteriori informazioni sulla stanza `QueueManager`, consultare [“Stanza QueueManager del file mqm.ini”](#) a pagina 105.

- c) Creare un backup della directory dei dati del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando `tar`. Ad esempio, per eseguire il back up della directory di dati per un gestore code è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo punto (punto):

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- d) Determinare l'ubicazione della directory di log del gestore code visualizzando IBM MQ file di configurazione del gestore code `qm.ini`. Questo file si trova nella directory dei dati del gestore code. Per ulteriori informazioni sul file, consultare [“File di configurazione del gestore code, qm.ini”](#) a pagina 107.

La directory di log del gestore code è definita come il valore della chiave `LogPath` nella stanza `Log`. Per informazioni sulla sezione, consultare [“Stanza di log del file qm.ini”](#) a pagina 141.

- e) Creare un backup della directory di log del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando `tar`. Ad esempio, per eseguire il back up della directory di log per un gestore code, è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo punto (punto):

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- f) Creare un backup dei repository dei certificati utilizzati dal gestore code se non si trovano nella directory dei dati del gestore code. Verificare che sia il file del database delle chiavi che il file stash delle password siano sottoposti a backup. Per informazioni sul repository delle chiavi del gestore code, consultare [Il repository delle chiavi SSL/TLS](#) e [Individuazione del repository delle chiavi per un gestore code](#). Per informazioni sull'individuazione dell'archivio chiavi AMS se il gestore

code è configurato per utilizzare l'intercettazione MCA (Message Channel Agent) AMS, consultare [Intercettazione MCA \(Message Channel Agent\)](#).

- g) Il gestore code esistente non è più richiesto, quindi può essere eliminato. Tuttavia, laddove possibile, è necessario eliminare il gestore code esistente solo dopo che è stato ripristinato correttamente nel sistema di destinazione. Il differimento dell'eliminazione garantisce che il gestore code possa essere riavviato se il processo di migrazione non viene completato correttamente.

Nota: Se si rinvia l'eliminazione del gestore code esistente, non riavviarla. È importante che il gestore code rimanga chiuso perché ulteriori modifiche alla relativa configurazione o stato vengono perse durante la migrazione.

2. Preparare il nodo RDQM primario:

- Creare un nuovo gestore code RDQM con lo stesso nome del gestore code di cui è stato eseguito il backup. Assicurarsi che il file system assegnato per il gestore code RDQM da **crtmqm** sia abbastanza grande da contenere i dati, i log primari e i log secondari per il gestore code esistente, oltre a spazio aggiuntivo per l'espansione futura. Per informazioni su come creare un gestore code RDQM, consultare [“Creazione di un RDQM HA”](#) a pagina 597.
- Determinare il nodo RDQM primario per il gestore code. Per informazioni su come determinare il nodo primario, vedere [rdqmstatus](#) (visualizzazione dello stato RDQM).
- Sul nodo RDQM primario, se il gestore code RDQM è avviato, arrestarlo utilizzando il comando `endmqm -w` o `endmqm -i`.
- Sul nodo RDQM primario, stabilire l'ubicazione delle directory di dati e di log per il gestore code RDQM (utilizzare i metodi descritti nei passi 1b e 1d).
- Sul nodo RDQM primario, eliminare il contenuto dei dati del gestore code RDQM e le directory di log, ma non le directory stesse.

3. Ripristinare il gestore code sul nodo RDQM primario:

- Copiare i backup dei dati del gestore code e le directory di log nel nodo RDQM primario, oltre a eventuali backup separati dei repository dei certificati utilizzati dal gestore code.
- Ripristinare il backup della directory di dati del gestore code nella directory di dati vuota per il nuovo gestore code RDQM, assicurando che la proprietà e le autorizzazioni del file siano conservate. Se il backup è stato creato utilizzando il comando `tar` di esempio nel passaggio 1c, l'utente root può utilizzare il seguente comando per ripristinarlo:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- Ripristinare il backup della directory di log del gestore code nella directory di log vuota per il nuovo gestore code RDQM, verificando che le autorizzazioni e la proprietà del file siano conservate. Se il backup è stato creato utilizzando il comando `tar` di esempio nel passo 1e, l'utente root può utilizzare il seguente comando per ripristinarlo:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- Modificare il file di configurazione del gestore code ripristinato, `qm.ini`, nella directory dei dati per il gestore code RDQM. Aggiornare il valore della chiave `LogPath` nella stanza `Log` per specificare la directory di log per il gestore code RDQM.

Esaminare gli altri percorsi file definiti nel file di configurazione e aggiornarli, se necessario. Ad esempio, potrebbe essere necessario aggiornare i seguenti percorsi:

- Il percorso per i file di log degli errori generati dai servizi di messaggi diagnostici.
 - Il percorso per le uscite richieste dal gestore code.
 - Il percorso per i file di caricamento switch se il gestore code è un coordinatore della transazione XA.
- Se il gestore code è configurato per utilizzare l'intercettazione AMS Message Channel Agent (MCA), copiare il keystore AMS nella nuova installazione RDQM, quindi esaminare e aggiornare la

configurazione. Il keystore deve essere disponibile su ciascun nodo RDQM, quindi se non è ubicato nel filesystem replicato per il gestore code, deve essere copiato su ciascun nodo. Per ulteriori informazioni, consultare [Intercettazione MCA \(Message Channel Agent\)](#).

- f) Verificare che il Gestore code sia visualizzato dal comando **dspmq** e che il suo stato sia riportato come terminato. Il seguente esempio mostra l'output di esempio per un gestore code HA RDQM:

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

- g) Verificare che i dati del gestore code ripristinati siano stati replicati sui nodi RDQM secondari utilizzando il comando **rdqmstatus** per visualizzare lo stato per il gestore code. Lo stato HA deve essere riportato come `Normal` su ciascun nodo. Il seguente esempio mostra l'output di esempio per un gestore code HA RDQM:

```
$ rdqmstatus -m QM1
Node: mqhavam10-adm
Queue manager status: Ended normally
Queue manager file system: 50MB used, 0.2GB allocated [42%]
HA role: Primary
HA status: Normal
HA control: Disabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None

Node: mqhavam11-adm
HA status: Normal

Node: mqhavam12-adm
HA status: Normal
```

- h) Avviare il gestore code sul nodo RDQM primario.
- i) Connettersi al gestore code e aggiornare il valore dell'attributo del gestore code `SSLKEYR` per specificare la nuova ubicazione del repository dei certificati del gestore code. Per impostazione predefinita, il valore di questo attributo è impostato su `queue_manager_data_directory/ssl/key`. Il repository certificati deve essere ubicato nella stessa ubicazione su ciascun nodo RDQM. Se il repository non si trova nel file system replicato per il gestore code, deve essere copiato su ciascun nodo.
- j) Esaminare le definizioni degli oggetti IBM MQ per il gestore code e aggiornare il valore degli attributi degli oggetti che fanno riferimento alle impostazioni di rete modificate, la directory di installazione di IBM MQ o la directory dei dati del gestore code, inclusi i seguenti oggetti:
- Indirizzi IP locali utilizzati dai listener (attributo `IPADDR`).
 - Indirizzi IP locali utilizzati dai canali (attributo `LOCLADDR`).
 - Indirizzi IP locali definiti per i canali riceventi del cluster (attributo `CONNAME`).
 - Gli indirizzi IP locali definiti per gli oggetti delle informazioni di comunicazione (attributo `GRPADDR`).
 - Percorsi di sistema definiti per le definizioni di processo e oggetto servizio.
- k) Arrestare e riavviare il gestore code per rendere effettive le modifiche.
- l) Ripetere il passo 3j per gestori code remoti, più le impostazioni equivalenti per le applicazioni, che si connettono al gestore code migrato, incluso:
- Nomi connessione canale (attributo `CONNAME`).
 - Le regole di autenticazione del canale che limitano le connessioni in entrata dal gestore code in base al relativo nome host o indirizzo IP.
 - Tabelle CCDT (Client Channel Definition Table), DNS (Domain Name Settings), instradamento di rete o informazioni di connessione equivalenti.

- m) Eseguire un failover gestito del gestore code su ciascun nodo RDQM per assicurarsi che la configurazione richiesta sia stata stabilita correttamente, fare riferimento a [“Impostazione della posizione preferita per un RDQM”](#) a pagina 605.

Ridimensionamento del file system per un gestore code HA RDQM

Per ridimensionare il filesystem per un gestore code di dati replicati (RDQM) HA (High Availability) esistente, eseguire il backup dei propri dati persistenti, quindi ripristinare i dati su un gestore code RDQM appena creato che ha lo stesso nome ma un filesystem di dimensione diversa.

Informazioni su questa attività

I gestori code di dati replicati HA richiedono un volume logico dedicato (filesystem) e la configurazione della replica del disco e il controllo HA. Questi componenti vengono configurati solo quando viene creato un nuovo gestore code. Il file system non può essere ridimensionato dopo che è stato creato perché deve avere la stessa dimensione su ogni nodo. Per ridimensionare il filesystem per un gestore code di dati replicati esistente (RDQM), è possibile eseguire il backup dei dati persistenti, quindi ripristinare i dati su un gestore code RDQM appena creato che ha lo stesso nome ma un filesystem di dimensione diversa. Questa procedura conserva la configurazione del gestore code, lo stato e i messaggi persistenti al momento della creazione del backup.

Procedura

1. Eseguire il backup del gestore code RDQM esistente sul nodo RDQM primario:

- Determinare il nodo RDQM primario per il gestore code. Per informazioni su come determinare il nodo primario, vedere [rdqmstatus](#) (visualizzazione dello stato RDQM) .
- Sul nodo RDQM primario, se il gestore code RDQM è avviato, arrestarlo utilizzando il comando **endmqm -w** o **endmqm -i** .
- Determinare l'ubicazione della directory dei dati del gestore code visualizzando il file di configurazione IBM MQ , `mqs.ini`. Su Linux, questo file si trova nella directory `/var/mqm` . Per ulteriori informazioni su `mqs.ini`, consultare [“File di configurazione IBM MQ , mqs.ini”](#) a pagina 94.

Individuare la sezione `QueueManager` per il gestore code nel file. La directory dei dati del gestore code è il valore della chiave denominata `DataPath`. Per ulteriori informazioni sulla stanza `QueueManager` , consultare [“Stanza QueueManager del file mqs.ini”](#) a pagina 105.

- Creare un backup della directory dei dati del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando **tar** . Ad esempio, per eseguire il back up della directory di dati per un gestore code è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo carattere punto (.):

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- Determinare l'ubicazione della directory di log del gestore code visualizzando IBM MQ file di configurazione del gestore code `qm.ini`. Questo file si trova nella directory dei dati del gestore code. Per ulteriori informazioni sul file, consultare [“File di configurazione del gestore code, qm.ini”](#) a pagina 107.

La directory di log del gestore code è definita come il valore della chiave `LogPath` nella stanza `Log`. Per informazioni sulla sezione, consultare [“Stanza di log del file qm.ini”](#) a pagina 141.

- Creare un backup della directory di log del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando **tar** . Ad esempio, per eseguire il back up della directory di log per un gestore code, è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo carattere punto (.):

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- Eliminare il gestore code RDQM esistente.

2. Ripristinare il gestore code con un filesystem della dimensione richiesta:

- a) Creare un nuovo gestore code RDQM con lo stesso nome del gestore code di cui è stato eseguito il backup. Assicurarsi che il file system assegnato per il gestore code RDQM da **crtmqm** sia la dimensione richiesta e che sia sufficientemente grande per contenere i dati, i log primari e i log secondari per il gestore code esistente, più dello spazio aggiuntivo per l'espansione futura. Per informazioni su come creare un gestore code RDQM, consultare [“Creazione di un RDQM HA” a pagina 597](#).
- b) Determinare il nodo RDQM primario per il gestore code. Per informazioni su come determinare il nodo primario, vedere [rdqmstatus \(visualizzazione dello stato RDQM\)](#).
- c) Sul nodo RDQM primario, se il gestore code RDQM è avviato, arrestarlo utilizzando il comando **endmqm -w** o **endmqm -i**.
- d) Sul nodo RDQM primario, stabilire la nuova ubicazione delle directory di dati e di log per il gestore code RDQM (utilizzare i metodi descritti ai passi 1c e 1e).
- e) Sul nodo RDQM primario, eliminare il contenuto dei dati del gestore code RDQM e le directory di log, ma non le directory stesse.
- f) Sul nodo RDQM primario, ripristinare il backup della directory di dati del gestore code nella directory di dati vuota per il nuovo gestore code RDQM, verificando che la proprietà e le autorizzazioni del file siano conservate. Se il backup è stato creato utilizzando il comando **tar** di esempio nel passo 1d, il seguente comando può essere utilizzato dall'utente root per ripristinarlo:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- g) Sul nodo RDQM principale, ripristinare il backup della directory di log del gestore code nella directory di log vuota per il nuovo gestore code RDQM, verificando che la proprietà del file e le autorizzazioni siano conservate. Se il backup è stato creato utilizzando il comando **tar** di esempio nel passo 1f, il seguente comando può essere utilizzato dall'utente root per ripristinarlo:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- h) Sul nodo RDQM primario, modificare il file di configurazione del gestore code ripristinato, **qm.ini**, nella directory dei dati per il nuovo gestore code RDQM. Aggiornare il valore della chiave **LogPath** nella stanza **Log** per specificare la directory di log per il nuovo gestore code RDQM determinato al passo 2d. Esaminare gli altri percorsi file definiti nel file di configurazione e aggiornarli, se necessario. Ad esempio, potrebbe essere necessario aggiornare i seguenti percorsi:

- Il percorso per i file di log degli errori generati dai servizi di messaggi diagnostici.
- Il percorso per le uscite richieste dal gestore code.
- Il percorso per i file di caricamento switch se il gestore code è un coordinatore della transazione XA.

- i) Verificare che il Gestore code sia visualizzato dal comando **dspmq** e che il suo stato sia riportato come terminato. Il seguente esempio mostra l'output di esempio per un gestore code HA RDQM:

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

- j) Verificare che i dati del gestore code ripristinati siano stati replicati sui nodi RDQM secondari utilizzando il comando **rdqmstatus** per visualizzare lo stato per il gestore code. Lo stato HA deve essere riportato come **Normal** su ciascun nodo. Il seguente esempio mostra l'output di esempio per un gestore code HA RDQM:

```
$ rdqmstatus -m QM1
Node: mqhvm10-adm
Queue manager status:      Ended normally
Queue manager file system: 50MB used, 0.2GB
allocated [42%]
HA role:                   Primary
HA status:                 Normal
HA control:                Disabled
HA current location:      This node
HA preferred location:    This node
```

```

HA floating IP interface:  None
HA floating IP address:  None
Node:                    mqhavm11-adm
HA status:               Normal
Node:                    mqhavm12-adm
HA status:               Normal

```

- k) Avviare il gestore code sul nodo RDQM primario.
- l) Eseguire un failover gestito del gestore code su ciascun nodo RDQM per assicurarsi che la configurazione richiesta sia stata stabilita correttamente, fare riferimento a [“Impostazione della posizione preferita per un RDQM”](#) a pagina 605.

Archiviazione dello stato dell'applicazione persistente

È possibile memorizzare le informazioni sullo stato persistente relative alle applicazioni insieme ad altri dati del gestore code.

Ogni gestore code IBM MQ dispone di un filesystem dedicato per il relativo stato persistente, che include sia i dati della coda che il log di recupero. In una configurazione RDQM, il filesystem è supportato da un volume logico replicato tra i sistemi Linux (nodi). Il file system include una directory `userdata` che è possibile utilizzare per memorizzare le informazioni sullo stato persistente per le applicazioni. Quindi, quando un gestore code di dati replicati si sposta per essere eseguito su un altro nodo nella configurazione RDQM, è disponibile il contesto dell'applicazione e il contesto del gestore code. Vedere [Contenuto della directory su Unix e Linux Systems](#).

Se si sceglie di memorizzare lo stato dell'applicazione nella directory `userdata`, è necessario tenere presente che i dati scritti in questa ubicazione potrebbero consumare lo spazio su disco disponibile assegnato al gestore code. È necessario assicurarsi che sia disponibile spazio su disco sufficiente per il gestore code per scrivere i dati della coda, i log e altre informazioni sullo stato persistente.

La directory `userdata` ha la proprietà utente e gruppo `mqm` ed è leggibile in modo che gli utenti possano accedervi senza dover far parte del gruppo di amministratori IBM MQ (`mqm`). Non è possibile modificare le autorizzazioni della directory `userdata`, ma è possibile creare il contenuto in essa con qualsiasi proprietà e autorizzazione richiesti.

Durante un failover del gestore code RDQM, il gestore code viene terminato e il relativo file system viene smontato sul nodo RDQM corrente. Il filesystem viene quindi montato e il gestore code viene riavviato su un altro nodo nella configurazione RDQM. Un file system non può essere smontato se un processo ha un handle aperto per uno dei suoi file. Per garantire che un failover del gestore code possa essere completato, se il filesystem del gestore code non può essere smontato, ai processi che hanno un handle di file aperto viene inviato un segnale `SIGTERM`, seguito da un `SIGKILL` se gli handle aperti non vengono rilasciati. Le applicazioni devono essere progettate per rispondere correttamente a `SIGTERM`. Se le applicazioni o i processi sono configurati come un servizio del gestore code, durante un failover gestito possono essere terminati durante l'arresto del gestore code prima che il filesystem venga smontato. Se un'applicazione o un processo non è configurato come servizio del gestore code o si verifica un failover non gestito, ad esempio una perdita del quorum, è probabile che i segnali vengano inviati per rilasciare il file system.

Linux *Impostazione della posizione preferita per un RDQM*

L'ubicazione preferita per un gestore code di dati replicati (RDQM) identifica il nodo in cui deve essere eseguito RDQM se tale nodo è disponibile.

Informazioni su questa attività

L'ubicazione preferita è il nome del nodo su cui Pacemaker deve eseguire il gestore code quando il gruppo HA è in uno stato normale (tutti i nodi e le connessioni disponibili). L'ubicazione preferita viene inizializzata con il nome del nodo principale quando viene creato il gestore code. È possibile eseguire i comandi per impostare l'ubicazione preferita su uno qualsiasi dei tre nodi. È necessario essere un utente che appartiene ai gruppi `mqm` e `haclient`.

Procedura

- Per assegnare il nodo locale o specificato come Ubicazione preferita per il gestore code denominato, immettere il seguente comando:

```
rdqmadm -p -m qmname [ -n nodename[,nodename ]
```

dove *qmname* è il nome dell'RDQM per cui si sta specificando l'ubicazione preferita e *nodename* è facoltativamente il nome del nodo preferito.

Se il gruppo HA si trova in uno stato normale e l'ubicazione preferita non è il nodo primario corrente, il gestore code viene arrestato e riavviato nella nuova ubicazione preferita. È possibile specificare un elenco separato da virgole di due nomi nodo per assegnare una seconda preferenza di Ubicazione preferita.

- Per cancellare la posizione preferita in modo che il gestore code non ritorni automaticamente a un nodo quando viene ripristinato, immettere il seguente comando:

```
rdqmadm -p -m qmname -d
```

Riferimenti correlati

[rdqmadm \(gestione cluster gestore code dati replicati\)](#)

Linux

Creazione ed eliminazione di un indirizzo IP mobile

Un indirizzo IP mobile consente al client di utilizzare lo stesso indirizzo IP per un gestore code di dati replicati (RDQM) indipendentemente dal nodo nel gruppo HA su cui è in esecuzione. (L'utilizzo di un indirizzo IP mobile è facoltativo.)

Informazioni su questa attività

È possibile creare o eliminare un indirizzo IP mobile utilizzando il comando **rdqmint**. L'indirizzo mobile si collega a un'interfaccia fisica denominata sul nodo primario per RDQM. Se l'RDQM esegue il failover e un nodo differente diventa il nodo primario, l'IP mobile viene collegato a un'interfaccia con lo stesso nome sul nuovo primario. Le interfacce fisiche sui tre nodi devono appartenere alla stessa sottorete dell'indirizzo IP mobile. Il seguente diagramma illustra l'utilizzo di un indirizzo IP mobile.

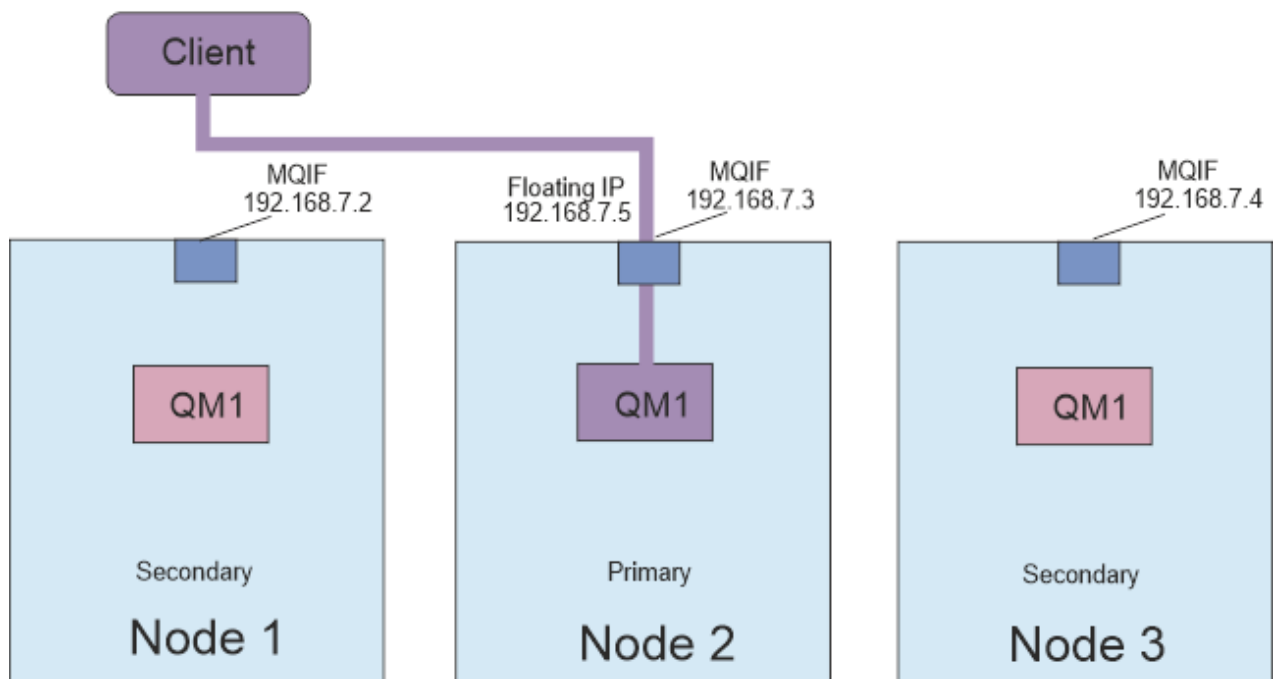
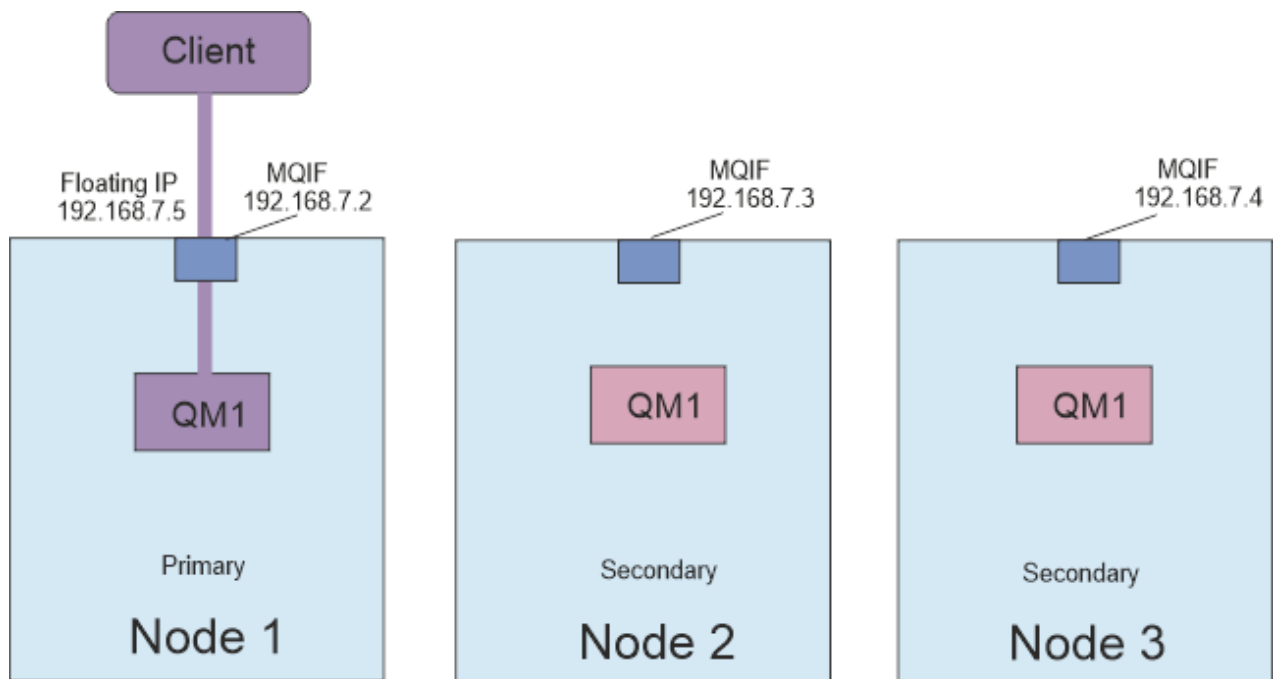


Figura 80. Indirizzo IP mobile

Per eseguire il comando `rdqmint`, è necessario essere un utente nei gruppi `mqm` e `haclient`. È possibile creare o eliminare l'indirizzo IP mobile sul nodo primario per RDQM o su uno dei nodi secondari.

Nota: Non è possibile utilizzare lo stesso indirizzo IP mobile per più RDQM, l'indirizzo IP mobile per ogni RDQM deve essere univoco.

Procedura

- Per creare un indirizzo IP mobile per un RDQM, immettere il seguente comando:

```
rdqmint -m qmname -a -f ipv4address -l interfacename
```

dove:

QMNAME

È il nome dell'RDQM per cui si sta creando l'indirizzo IP mobile.

ipv4address

L'indirizzo IP mobile in formato ipv4 .

L'indirizzo IP mobile deve essere un indirizzo IPv4 valido che non sia già definito su alcun nodo HA e deve appartenere alla stessa sottorete degli indirizzi IP statici definiti per l'interfaccia locale.

interfaceName

Il nome dell'interfaccia fisica sul nodo primario a cui collegarsi.

Ad esempio:

```
rdqmint -m QM1 -a -f 192.168.7.5 -l MQIF
```

- Per eliminare un indirizzo IP mobile esistente, immettere il comando seguente:

```
rdqmint -m qmname -d
```

Riferimenti correlati

[rdqmint \(aggiungere o eliminare l'indirizzo IP mobile per RDQM\)](#)

Linux

Avvio, arresto e visualizzazione dello stato di un RDQM HA

Utilizzare varianti di comandi di controllo IBM MQ standard per avviare, arrestare e visualizzare lo stato corrente di un gestore code di dati replicati (RDQM).

Informazioni su questa attività

È necessario eseguire i comandi che avviano, arrestano e visualizzano lo stato corrente di un gestore code di dati replicati (RDQM) come utente appartenente ai gruppi `mqm` e `haclient` .

È necessario eseguire i comandi per avviare e arrestare un gestore code sul nodo primario per tale gestore code.

Procedura

- Per avviare un RDQM, immettere il seguente comando sul nodo primario di RDQM:

```
strmqm qmname
```

dove `qmname` è il nome dell'RDQM che si desidera avviare.

RDQM viene avviato e Pacemaker avvia la gestione di RDQM. È necessario specificare l'opzione `-ns` con `strmqm` se si desidera specificare altre opzioni `strmqm` .

- Per arrestare un RDQM, immettere il seguente comando sul nodo primario del RDQM:

```
endmqm qmname
```

dove `qmname` è il nome dell'RDQM che si desidera arrestare.

Pacemaker cessa di gestire RDQM, quindi l'RDQM viene terminato. Tutti gli altri parametri `endmqm` possono essere utilizzati quando si arresta un RDQM.

- Per visualizzare lo stato di un RDQM, immettere il seguente comando:

```
dspmq
```

Le informazioni sullo stato che vengono emesse dipendono dal fatto che si esegua il comando sul nodo primario o secondario di RDQM. Se eseguito sul nodo primario, viene visualizzato uno dei

normali messaggi di stato restituiti da **dspmq** . Se si esegue il comando su un nodo secondario, viene visualizzato lo stato `running elsewhere` . Ad esempio, se **dspmq** viene eseguito sul nodo RDQM7, potrebbero essere restituite le seguenti informazioni:

```
QMNAME (RDQM8)           STATUS(Running elsewhere)
QMNAME (RDQM9)           STATUS(Running elsewhere)
QMNAME (RDQM7)           STATUS(Running)
```

Se il nodo primario non è disponibile o se **dspmq** viene eseguito da un utente non `root` o da un membro del gruppo `haclient` , viene riportato lo stato `Unavailable` . Ad esempio:

```
QMNAME (RDQM8)           STATUS(Unavailable)
QMNAME (RDQM9)           STATUS(Unavailable)
QMNAME (RDQM7)           STATUS(Unavailable)
```

È possibile immettere il comando **dspmq -o ha** (o **dspmq -o HA**) per visualizzare un elenco di gestori code noti a un nodo e se sono RDQM o meno, ad esempio:

```
dspmq -o ha

QMNAME (RDQM8)           HA (Replicated)
QMNAME (RDQM9)           HA (Replicated)
QMNAME (RDQM7)           HA (Replicated)
QMNAME (QM7)             HA ()
```

Riferimenti correlati

[dspmq](#) (visualizza gestori code)

[endmqm](#) (fine gestore code)

[strmqm](#) (avvio gestore code)

Azioni della risorsa non riuscite

Le azioni delle risorse non riuscite si verificano quando il componente Pacemaker di una configurazione ad alta disponibilità RDQM rileva alcuni problemi con una risorsa su uno dei nodi in un gruppo HA.

La soluzione RDQM HA utilizza Pacemaker per il monitoraggio e la gestione delle risorse (consultare [“Alta disponibilità RDQM”](#) a pagina 588). Se Pacemaker rileva un errore durante l'esecuzione di un'operazione su una risorsa su un nodo, registra queste informazioni utilizzando un'azione di risorsa non riuscita.

Alcune azioni della risorsa non riuscite impediscono l'esecuzione della risorsa e devono essere eliminate prima che Pacemaker possa riavviare la risorsa.

È possibile utilizzare il comando **rdqmstatus -m** per verificare se sono presenti azioni di risorse non riuscite che stanno arrestando l'avvio di un gestore code su uno o più nodi.

È quindi possibile utilizzare il comando **rdqmstatus -m qmname -a** per visualizzare i dettagli delle azioni delle risorse non riuscite associate ad un gestore code. Seguire questa azione utilizzando il comando **rdqmclean** per cancellare queste azioni della risorsa non riuscite e quindi liberare tutte le risorse limitate. (È inoltre necessario eseguire un'azione per risolvere i problemi che hanno causato l'azione della risorsa non riuscita.)

Le seguenti risorse sono controllate da Pacemaker in una configurazione HA RDQM e possono essere oggetto di azioni di risorse non riuscite:

- Gestore code
- IP mobile
- Controllo RDQM
- File system
- Replica DR (DRBD)
- Replica HA (DRBD)

Ogni tipo di risorsa può essere soggetto ai seguenti tipi di errore:

Soft

Gli errori lievi sono transitori e Pacemaker continua a tentare di recuperare la risorsa fino a quando non va in timeout o viene altrimenti arrestato.

Difficile

Un errore grave richiede un intervento amministrativo. Gli errori gravi bloccano l'esecuzione della risorsa su un nodo particolare.

Irreversibile

Un errore grave richiede un intervento di gestione. Gli errori irreversibili bloccano l'esecuzione della risorsa su qualsiasi nodo.

Consultare [“Visualizzazione dello stato del gruppo RDQM e HA” a pagina 610](#) per esempi di stato incluse le azioni della coda di risorse non riuscite.

È possibile utilizzare il comando **rdqmclean** per cancellare tutte le azioni di risorsa non riuscite associate ad un gestore code specificato o tutte le azioni di risorsa non riuscite nella configurazione RDQM HA.

Nota: Alcune azioni delle risorse non riuscite non determinano il blocco del gestore code su un nodo. Ad esempio, dopo una fine del gestore code non prevista, Pacemaker tenta di riavviare il gestore code sul nodo su cui è stato rilevato che non è in esecuzione. Se l'avvio ha esito positivo, l'esecuzione del gestore code non viene bloccata sul nodo. L'unico modo per essere consapevoli dell'azione della risorsa non riuscita in questo caso è eseguendo **rdqmstatus -m qmname -a**.

Attività correlate

[“Visualizzazione dello stato del gruppo RDQM e HA” a pagina 610](#)

È possibile visualizzare lo stato del gruppo HA e dei singoli gestori code di dati replicati (RDQM).

Riferimenti correlati

[rdqmclean](#)

[rdqmstatus](#)

Linux

Visualizzazione dello stato del gruppo RDQM e HA

È possibile visualizzare lo stato del gruppo HA e dei singoli gestori code di dati replicati (RDQM).

Informazioni su questa attività

Utilizzare il comando **rdqmstatus** per visualizzare lo stato di singoli RDQM e del gruppo HA nel suo complesso.

Lo stato di riepilogo per un nodo visualizza anche le informazioni sul modulo kernel DRBD su cui si basa RDQM. Quando si aggiorna RDQM, è importante assicurarsi che la versione corretta del modulo kernel DRBD sia installata per la versione del kernel RHEL in esecuzione sul sistema. Lo stato visualizza la versione del kernel del sistema operativo, la versione del kernel per cui è stato creato il modulo DRBD, la versione DRBD e lo stato caricato del modulo del kernel DRBD.

È necessario essere un utente nei gruppi `mqm` e `haclient` per eseguire il comando **rdqmstatus**. È possibile eseguire il comando su uno qualsiasi dei tre nodi.

Procedura

- Per visualizzare lo stato di riepilogo di un nodo e degli RDQM che fanno parte della configurazione HA:

```
rdqmstatus
```

Viene visualizzata l'identità del nodo su cui è stato eseguito il comando su, i dettagli del kernel e DRBD per tale nodo, e lo stato di RDQM nella configurazione HA, ad esempio:

```
Node:                               mqhavm07.exampleco.com
OS kernel version:                  5.14.0-362.18.1
DRBD OS kernel version:              5.14.0-362.18.1
DRBD version:                        9.2.7
DRBD kernel module status:           Loaded
```

```

Queue manager name:          RDQM8
Queue manager status:       Running elsewhere
HA current location:        mqhavam08.exampleco.com
HA preferred location:      mqhavam08.exampleco.com
HA blocked location:        None

Queue manager name:          RDQM9
Queue manager status:       Running elsewhere
HA current location:        mqhavam09.exampleco.com
HA preferred location:      mqhavam09.exampleco.com
HA blocked location:        None

Queue manager name:          RDQM7
Queue manager status:       Running
HA current location:        This node
HA preferred location:      This node
HA blocked location:        None

```

Lo stato del modulo kernel DRBD è uno dei seguenti valori:

Caricato

Indica che il modulo DRBD è stato caricato.

Caricato parzialmente

Può verificarsi quando il modulo DRBD è stato caricato, ma non funziona correttamente a causa di una mancata corrispondenza.

Non caricato

Il modulo DRBD non è caricato. Può essere visualizzato su una configurazione appena installata, quando non è stato ancora creato alcun gestore code RDQM.

Non installato

Indica che il modulo DRBD non è installato. oppure che IBM MQ non è stato in grado di determinare la versione kernel del sistema operativo del modulo DRBD.

La versione precedentemente installata è ancora caricata

Questo stato può verificarsi se un nuovo modulo DRBD è installato mentre il modulo DRBD esistente è in esecuzione (ossia, un gestore code RDQM è in esecuzione). Il modulo appena installato viene riportato nello stato, ma non è il modulo attualmente in esecuzione.

- Per visualizzare lo stato dei tre nodi nel gruppo HA, immettere il seguente comando:

```
rdqmstatus -n
```

Viene riportato lo stato in linea o fuori linea di ciascun nodo. Ad esempio:

```

Node mqha04(mqhavam04.example.com) is online
Node mqha05(mqhavam05.example.com) is offline
Node mqha06(mqhavam06.example.com) is online

```

- Per visualizzare lo stato di un particolare gestore code su tutti i nodi nel gruppo HA, immettere il seguente comando:

```
rdqmstatus -m qmname
```

dove *qmname* è il nome dell'RDQM per cui si desidera visualizzare lo stato. Viene visualizzato lo stato di RDQM sul nodo corrente, seguito da un riepilogo dello stato degli altri due nodi dalla prospettiva del nodo corrente.

- Per visualizzare lo stato di un particolare gestore code su tutti i nodi nel gruppo HA, inclusi i dettagli di eventuali azioni di risorse non riuscite, immettere il seguente comando:

```
rdqmstatus -m qmname -a
```

dove *qmname* è il nome dell'RDQM per cui si desidera visualizzare lo stato. Viene visualizzato lo stato di RDQM sul nodo corrente, seguito da un riepilogo dello stato degli altri due nodi dalla prospettiva del nodo corrente. Questo è seguito dai dettagli relativi a tutte le azioni di risorsa non riuscite associate a RDQM.

•

La seguente tabella riepiloga le informazioni sul nodo corrente che possono essere restituite dal comando `rdqmstatus -m qmname` per un RDQM.

<i>Tabella 33. Stato nodo corrente</i>		
Attributo Stato	Valori possibili	Quando visualizzato
Nome nodo	<i>NODENAME</i>	Sempre visualizzato
Stato gestore code	In esecuzione In esecuzione altrove Terminato Non disponibile	Sempre visualizzato
CPU	<i>n.nn%</i>	Viene visualizzato solo quando il nodo corrente ha un ruolo primario (ossia, RDQM è in esecuzione su questo nodo)
Memoria	<i>nnn</i> MB utilizzati, <i>y.y</i> GB assegnati	Viene visualizzato solo quando il nodo corrente ha un ruolo primario (ossia, RDQM è in esecuzione su questo nodo)
File system del gestore code	<i>nnn</i> MB utilizzati, <i>y.y</i> GB assegnati [z%]	Viene visualizzato solo quando il nodo corrente ha un ruolo primario (ossia, RDQM è in esecuzione su questo nodo)
Ruolo HA	Secondario primario sconosciuto	Sempre visualizzato
Stato HA	Tutti i nodi in standby Questo nodo è in standby Nodi remoti in standby Misto <i>stato dei nodi remoti</i>	Tutti i nodi in standby Nodo corrente in standby Entrambi i nodi remoti in standby Stato differente per ogni nodo remoto (consultare la tabella successiva per lo stato individuale) Stesso stato per entrambi i nodi remoti (consultare la tabella successiva per tutti i valori)
Controllo HA	Abilitato/a Disabilitato/a Sconosciuto	Sempre visualizzato. Mostra se RDQM è sotto il controllo Pacemaker
Ubicazione preferita HA	Nessuna Questo nodo Sconosciuto <i>NODENAME</i>	Sempre visualizzato

Tabella 33. Stato nodo corrente (Continua)

Attributo Stato	Valori possibili	Quando visualizzato
Ubicazione bloccata HA	Nessuno - L'esecuzione del gestore code non è bloccata su alcun nodo Questo nodo - Il gestore code è bloccato dall'esecuzione sul nodo corrente a causa di una o più azioni di risorsa non riuscite <i>nodename</i> - L'esecuzione del gestore code è bloccata su <i>nodename</i> a causa di una o più azioni di risorsa non riuscite <i>nodename1, nodename2</i> - L'esecuzione del gestore code è bloccata su <i>nodename1</i> e <i>nodename2</i> a causa di una o più azioni della risorsa non riuscite Tutti i nodi - L'esecuzione del gestore code è bloccata su tutti i nodi a causa di una o più azioni di risorse non riuscite	Sempre visualizzato
Interfaccia IP mobile HA	<i>nome_interfaccia</i>	Sempre visualizzato
Indirizzo IP mobile HA	<i>IPV4_address</i>	Sempre visualizzato

La seguente tabella riepiloga le informazioni restituite dal comando `rdqmstatus -m qmname` per gli altri nodi nel gruppo HA.

Tabella 34. Stato altro nodo

Attributo Stato	Valori possibili	Quando visualizzato
Nome nodo	<i>nodename</i>	Sempre visualizzato
Stato HA	Normale Sincronizzazione in corso Remoto non disponibile Non congruente In pausa Nodo remoto in standby Sconosciuto	I nodi sono sincronizzati tra loro Sincronizzazione con il nodo remoto Impossibile comunicare con il nodo remoto Non sincronizzato con il nodo remoto e non sincronizzato Replica sospesa Nodo remoto in standby
Sincronizzazione HA in corso	<i>n.n%</i>	Visualizzato quando la sincronizzazione è in corso e il comando viene eseguito come root
Tempo di sincronizzazione stimato HA	<i>aaaa - mm - gg hh:mm:ss.nnn</i>	Visualizzato quando la sincronizzazione è in corso
Dati HA non sincronizzati	<i>nKB</i>	Visualizzato quando il nodo remoto non è disponibile o non è congruente

Tabella 34. Stato altro nodo (Continua)

Attributo Stato	Valori possibili	Quando visualizzato
Ultima sincronizzazione HA	<i>aaaa - mm - gg hh:mm:ss.nnn</i>	Visualizzato quando i dati HA non sono sincronizzati (dopo la sincronizzazione iniziale). Fornisce l'ora e la data in cui i dati sono stati sincronizzati l'ultima volta.

Esempio

Esempio di stato normale sul nodo primario:

```

Node: mqhavam07.exampleco.com
Queue manager status: Running
CPU: 0.00
Memory: 123MB
Queue manager file system: 606MB used, 1.0GB allocated [60%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: eth4
HA floating IP address: 192.0.2.4

Node: mqhavam08.exampleco.com
HA status: Normal

Node: mqhavam09.exampleco.com
HA status: Normal

```

Esempio di stato normale su un nodo secondario:

```

Node: mqhavam08.exampleco.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavam07.exampleco.com
HA preferred location: mqhavam07.exampleco.com
HA blocked location: None
HA floating IP interface: eth4
HA floating IP address: 192.0.2.4

Node: mqhavam07.exampleco.com
HA status: Normal

Node: mqhavam09.exampleco.com
HA status: Normal

```

Esempio di stato sul nodo primario quando è in corso la sincronizzazione:

```

Node: mqhavam07.exampleco.com
Queue manager status: Running
CPU: 0.53
Memory: 124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Synchronization in progress
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: eth4
HA floating IP address: 192.0.2.4

Node: mqhavam08.exampleco.com
HA status: Synchronization in progress

```

```

HA synchronization progress:      11.0%
HA estimated time to completion:  2017-09-06 14:55:05

Node:                             mqhavam09.exampleco.com
HA status:                        Synchronization in progress
HA synchronization progress:      11.0%
HA estimated time to completion:  2017-09-06 14:55:06

```

Esempio di stato sul nodo primario quando la sincronizzazione è stata persa:

```

Node:                             mqhavam07.exampleco.com
Queue manager status:             Running
CPU:                             0.53
Memory:                          124MB
Queue manager file system:       51MB used, 1.0GB allocated [5%]
HA role:                         Primary
HA status:                       Mixed
HA control:                      Enabled
HA current location:             This node
HA preferred location:          This node
HA blocked location:            None
HA floating IP interface:       eth4
HA floating IP address:         192.0.2.4

Node:                             mqhavam08.exampleco.com
HA status:                       Normal

Node:                             mqhavam09.exampleco.com
HA status:                       Inconsistent
HA out of sync data:            15932KB
HA last in sync:               2017-09-06 14:55:06

```

Esempio di un nodo principale che mostra più stati:

```

Node:                             mqhavam07.exampleco.com
Queue manager status:             Running
CPU:                             0.02
Memory:                          124MB
Queue manager file system:       51MB used, 1.0GB allocated [5%]
HA role:                         Primary
HA status:                       Mixed
HA control:                      Enabled
HA current location:             This node
HA preferred location:          This node
HA blocked location:            None
HA floating IP interface:       eth4
HA floating IP address:         192.0.2.4

Node:                             mqhavam08.exampleco.com
HA status:                       Normal

Node:                             mqhavam09.exampleco.com
HA status:                       Inconsistent

```

Esempio di un nodo primario che mostra le azioni delle risorse non riuscite:

```

Node:                             mqhavam07.exampleco.com
Queue manager status:             Running
CPU:                             0.00%
Memory:                          123MB
Queue manager file system:       606MB used, 1.0GB allocated [60%]
HA role:                         Primary
HA status:                       Normal
HA control:                      Enabled
HA current location:             This node
HA preferred location:          mqhavam08.exampleco.com
HA blocked location:            mqhavam08.exampleco.com
HA floating IP interface:       eth4
HA floating IP address:         192.0.2.4

Node:                             mqhavam08.exampleco.com
HA status:                       Normal

Node:                             mqhavam09.exampleco.com
HA status:                       Normal

Failed resource action:          Start
Resource type:                  Filesystem

```

```

Failure node:                mqhavam08.exampleco.com
Failure time:                2017-09-06 12:00:00
Failure reason:             Couldn't find directory [/var/mqm/vols/qmname] to use
as a mount point
Blocked location:          mqhavam08.exampleco.com

```

Questo stato mostra che Pacemaker non è stato in grado di avviare il filesystem sul nodo mqhavam08.exampleco.com alle 12:00. Questa azione di risorsa non riuscita indica che l'esecuzione del gestore code è bloccata su mqhavam08.exampleco.com. Una volta risolto il problema sottostante che ha causato l'azione della risorsa non riuscita, eseguire il comando **rdqmclean** per cancellare l'azione non riuscita in modo che Pacemaker possa ritentare l'azione (se richiesto).

Esempio di uno stato di riepilogo che mostra una mancata corrispondenza tra la versione kernel del sistema operativo (RHEL 9.3) e il modulo kernel DRBD (destinato a RHEL 9.2). Anche se lo stato riporta che il modulo del kernel DRBD è caricato e il gestore code è in esecuzione, è necessario aggiornare il modulo del kernel DRBD con la versione destinata al kernel del sistema operativo in esecuzione in questa situazione.

```

Node:                        mqhavam07.exampleco.com
OS kernel version:          5.14.0-362.18.1
DRBD OS kernel version:    5.14.0-284.11.1
DRBD version:              9.2.7+ptf.14
DRBD kernel module status:  Loaded

Queue manager name:        RDQM7
Queue manager status:      Running
HA current location:       This node
HA preferred location:     This node
HA blocked location:       None

```

Esempio di stato di riepilogo che mostra una mancata corrispondenza tra la versione del kernel del sistema operativo (RHEL 8.10) e il modulo del kernel DRBD (destinato a RHEL 8.8). In questo esempio la mancata corrispondenza della versione è più grave e il caricamento del modulo del kernel DRBD non riesce. Di conseguenza, il gestore code non riesce ad avviare il nodo preferito e il relativo stato HA in Unknown. Per risolvere questo errore, il modulo kernel DRBD deve essere aggiornato con la destinazione della versione per il kernel del sistema operativo in esecuzione.

```

Node:                        mqhavam57.exampleco.com
OS kernel version:          4.18.0-553
DRBD OS kernel version:    4.18.0-477
DRBD version:              9.2.7+ptf.14
DRBD kernel module status:  Partially loaded

Queue manager name:        QM2
Queue manager status:      Running elsewhere
HA status:                 Unknown
HA current location:       mqhavam58.exampleco.com
HA preferred location:     This node
HA blocked location:       All nodes

```

Riferimenti correlati

 [rdqmstatus](#)

Modifica degli indirizzi IP nelle configurazioni ad alta disponibilità

Se si modificano gli indirizzi IP di una qualsiasi delle interfacce in una configurazione ad alta disponibilità, l'operazione di alta disponibilità non sarà più disponibile e il gestore code non verrà eseguito sul nodo in cui sono stati modificati gli indirizzi.

Specificare fino a tre indirizzi IP per l'operazione HA nel file `rdqm.ini`. Se gli indirizzi del monitor Pacemaker sono già stati modificati, è necessario ripristinarli temporaneamente ai valori originali prima di seguire la procedura. Altrimenti non è possibile eliminare il gestore code HA RDQM.

1. Rimuovere la configurazione HA su ciascun nodo. Rimuovere l'HA eseguendo il backup dei gestori code e quindi eliminandoli, consultare “Backup e ripristino dei dati del gestore code IBM MQ” a pagina 696 e “Eliminazione di un RDQM HA” a pagina 599, quindi rimuovere il gruppo HA stesso, consultare “Eliminazione del cluster Pacemaker (gruppo HA)” a pagina 597.

2. Ricreare la configurazione HA con i nuovi indirizzi IP, consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593.
3. Creare di nuovo i gestori code HA e ripristinare il backup, consultare [“Creazione di un RDQM HA”](#) a pagina 597 e [“Backup e ripristino dei dati del gestore code IBM MQ”](#) a pagina 696.

Linux **Sostituzione di un nodo malfunzionante in una configurazione ad alta disponibilità**

Se uno dei nodi nel tuo gruppo HA ha esito negativo, puoi sostituirlo.

Informazioni su questa attività

Le operazioni da eseguire per sostituire un nodo dipendono dallo scenario:

- Se si sta sostituendo il nodo malfunzionante con un nodo con una configurazione identica, è possibile sostituire il nodo senza interrompere il gruppo HA.
- Se il nuovo nodo ha una configurazione differente, è necessario eliminare e quindi ricreare il gruppo HA. È possibile prima eseguire il backup dei gestori code dal nodo su cui sono in esecuzione, quindi ripristinarli dopo aver rigenerato il gruppo HA.

Procedura

- Se il nodo di sostituzione è configurato per essere simile al nodo non riuscito (stesso nome host, stessi indirizzi IP e così via), completare la seguente procedura sul nuovo nodo:
 - a) Creare un file `rdqm.ini` che corrisponda ai file sugli altri nodi, quindi eseguire il comando `rdqmadm -c` (consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593).
 - b) Eseguire il comando `crtmqm -sxs qmanager` per creare nuovamente ciascun gestore code di dati replicati (consultare [“Creazione di un RDQM HA”](#) a pagina 597).
- Se il nodo di sostituzione ha una configurazione diversa rispetto al nodo in errore:
 - a) Se necessario, eseguire il backup dei gestori code (consultare [“Backup e ripristino dei dati del gestore code IBM MQ”](#) a pagina 696).
 - b) Eliminare i gestori code di dati replicati dagli altri nodi nel gruppo HA utilizzando il comando `dltmqm` (consultare [“Eliminazione di un RDQM HA”](#) a pagina 599).
 - c) Annullare la configurazione del cluster Pacemaker utilizzando il comando `rdqmadm -u` (consultare [“Eliminazione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 597).
 - d) Riconfigurare il cluster Pacemaker, incluse le informazioni per il nuovo nodo, utilizzando il comando `rdqmadm -c` (consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593).
 - e) Se necessario (ossia, se non si dispone dell'accesso SSH agli altri nodi), eseguire il comando `crtmqm -sxs qmanager` per ricreare ciascun gestore code di dati replicati sugli altri nodi (consultare [“Creazione di un RDQM HA”](#) a pagina 597).
 - f) Eseguire il comando `crtmqm -sx qmanager` per creare i gestori code sul nodo di sostituzione.
 - g) Se necessario, ripristinare i dati e la configurazione sui propri gestori code (consultare [“Backup e ripristino dei dati del gestore code IBM MQ”](#) a pagina 696).

MQ Adv. Linux **Ripristino di emergenza RDQM**

RDQM (gestore code di dati replicati) è disponibile su un sottoinsieme di piattaforme Linux e può fornire una soluzione di ripristino di emergenza.

Consultare [Software Product Compatibility Reports](#) per i dettagli completi.

È possibile creare un'istanza primaria di un gestore code di ripristino di emergenza in esecuzione su un server e un'istanza secondaria del gestore code su un altro server che funga da nodo di ripristino. I dati vengono replicati tra le istanze del gestore code. Se si perde il gestore code primario, è possibile creare manualmente l'istanza secondaria nell'istanza primaria e avviare il gestore code, quindi riprendere

il lavoro dallo stesso luogo. Non è possibile avviare un gestore code mentre è nel ruolo secondario. La replica dei dati tra i due nodi è gestita da DRBD.

È possibile scegliere tra la replica sincrona e asincrona dei dati tra gestori code primari e secondari. Se si seleziona l'opzione asincrona, le operazioni quali IBM MQ PUT o GET vengono completate e ritornano all'applicazione prima che l'evento venga replicato sul gestore code secondario. La replica asincrona indica che, in seguito a una situazione di recupero, alcuni dati di messaggistica potrebbero essere persi. Ma il gestore code secondario sarà in uno stato congruente e sarà in grado di avviare l'esecuzione immediatamente, anche se viene avviato in una parte leggermente precedente del flusso di messaggi.

Non è possibile aggiungere il ripristino di emergenza a un gestore code esistente, sebbene sia possibile migrare un gestore code esistente in modo che diventi un gestore code RDQM (consultare [“Migrazione di un gestore code per diventare un gestore code DR RDQM”](#) a pagina 625).

È possibile avere diverse coppie di gestori code RDQM in esecuzione su un numero di server differenti. Ad esempio, è possibile disporre di gestori code di ripristino di emergenza primari in esecuzione su nodi differenti, mentre tutti i gestori code di ripristino di emergenza secondari vengono eseguiti sullo stesso nodo. Alcune configurazioni di esempio sono illustrate nei seguenti diagrammi.

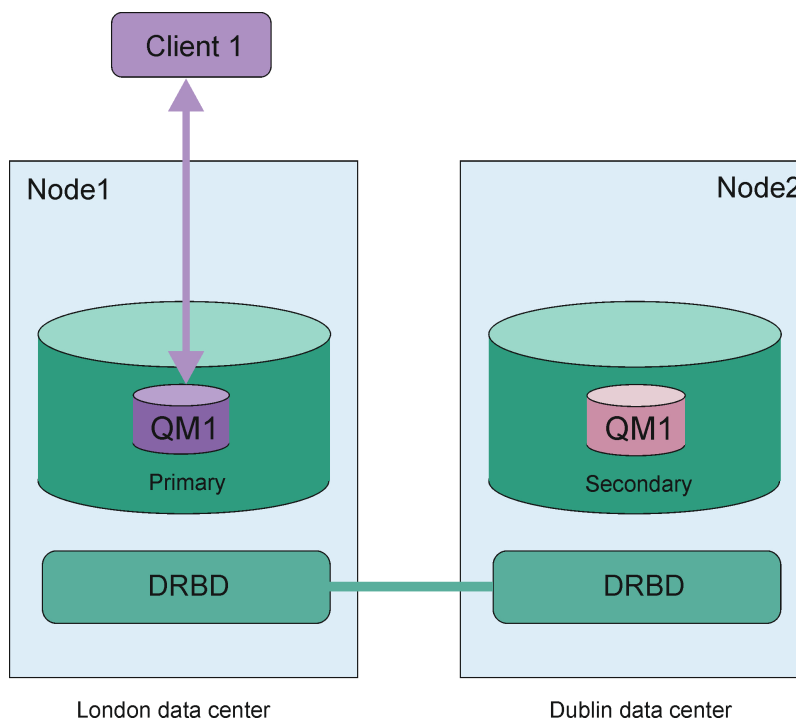


Figura 81. Singola coppia RDQM

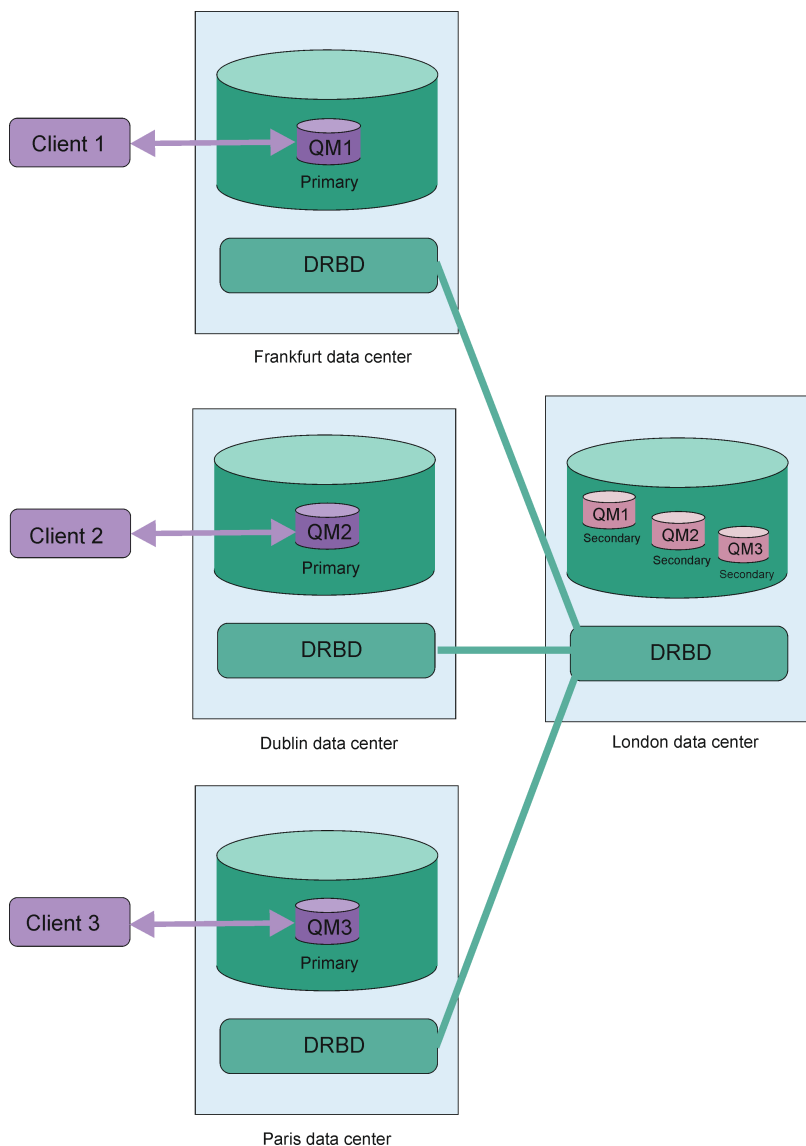


Figura 82. Gestori code secondari nello stesso nodo

Replica, sincronizzazione e istantanee

Mentre i due nodi in una configurazione di ripristino di emergenza sono connessi, tutti gli aggiornamenti ai dati persistenti per un gestore code di ripristino di emergenza vengono trasferiti dall'istanza primaria del gestore code all'istanza secondaria. Questa operazione è nota come **replica**.

Se la connessione di rete tra i due nodi viene persa, le modifiche ai dati persistenti per l'istanza primaria di un gestore code vengono tracciate. Quando la connessione di rete viene ripristinata, viene utilizzato un processo diverso per rendere l'istanza secondaria più rapida possibile. Questa operazione è nota come **sincronizzazione**.

Mentre la sincronizzazione è in esecuzione, i dati sull'istanza secondaria si trovano in uno stato incongruente. Viene eseguita un' **istantanea** dello stato dei dati del gestore code secondario. Se si verifica un errore del nodo principale o della connessione di rete durante la sincronizzazione, l'istanza secondaria ritorna a questa istantanea e il gestore code può essere avviato. Tuttavia, tutti gli aggiornamenti che si sono verificati dopo l'errore di rete originale vengono persi.

Dati partizionati (cervello diviso)

Le configurazioni DR RDQM richiedono l'intervento dell'utente dopo la perdita dell'istanza primaria di un gestore code per promuovere ed eseguire l'istanza secondaria sul nodo di recupero. È responsabilità di chiunque (o qualsiasi altra cosa) promuova l'istanza secondaria per garantire che il precedente gestore code primario sia arrestato. Se il primario originale continua ad essere in esecuzione, potrebbe elaborare i messaggi e, quando viene ripristinata la normale operazione, le due istanze del gestore code hanno viste differenti dei dati. Questo è noto come stato partizionato o split - brain.

Considera le seguenti situazioni:

- Il nodo su cui è in esecuzione il gestore code primario ha esito negativo. L'istanza secondaria viene promossa in modo che diventi l'istanza; non è possibile eseguire azioni per arrestare l'istanza primaria originale perché non è in esecuzione. Quando il nodo originale viene ripristinato o sostituito, il gestore code su tale nodo verrà inizialmente reso secondario e sincronizzato con il gestore code primario sul nodo di recupero. I ruoli dei due gestori code vengono quindi invertiti e le normali operazioni ricominciano. L'unica perdita di dati potenziale in questa situazione è qualsiasi dato che il primario non ha completato la replica sul secondario prima che il nodo non funzionasse.
- Si è verificato un errore di rete che interessa il link di replica tra i nodi che eseguono le istanze primarie e secondarie del gestore code. In questa situazione è necessario assicurarsi di arrestare il primario originale prima di promuovere il secondario. Se il primario originale dispone ancora di un'altra connettività di rete, si hanno effettivamente due istanze primarie in esecuzione contemporaneamente e i dati partizionati possono accumularsi. (Se il link di replica funziona, non è possibile promuovere un gestore code secondario se l'istanza primaria è ancora in esecuzione, il comando ha esito negativo.)
- Si è verificato un errore di rete completo sul nodo che esegue l'istanza primaria del gestore code. Ancora una volta, è necessario assicurarsi di arrestare l'istanza primaria prima di promuovere quella secondaria. Se il primario precedente è ancora in esecuzione quando la rete viene ripristinata, ci saranno due istanze primarie e di nuovo i dati partizionati si accumuleranno.

Quando si esegue un failover gestito, non dovrebbe essere visualizzato uno stato DR *partitioned* per le istanze del gestore code. Un failover gestito termina il gestore code sul nodo primario, quindi avvia il gestore code sul nodo di recupero dopo che i dati sono stati completamente replicati. Uno stato partizionato non è previsto perché il gestore code è terminato e i dati sono sincronizzati tra i nodi prima che vengano avviati sul nodo di ripristino. Se il gestore code viene avviato sul nodo di recupero mentre si verifica una perdita di connettività tra i nodi, è probabile che la divergenza dei dati sia dovuta al fatto che il gestore code era attivo sul nodo principale quando si è persa la connettività. In questo scenario, è previsto che venga riportato uno stato partizionato una volta ripristinata la connettività perché i dati del gestore code non sono stati sincronizzati. Se si verifica uno stato partizionato, potrebbe essere necessario esaminare i due dataset e prendere una decisione informata su quale serie conservare. Consultare [“Risoluzione di un problema partizionato \(split brain\) in DR RDQM”](#) a pagina 640.

Requisiti per la soluzione RDQM DR

È necessario soddisfare una serie di requisiti prima di configurare una coppia di gestori code DR (Disaster Recovery) RDQM.

Requisiti di sistema

Prima di configurare RDQM DR, è necessario completare alcune operazioni di configurazione su ciascuno dei server su cui si trovano i gestori code RDQM DR.

- Ogni nodo richiede un gruppo di volumi denominato `drbdpool`. La memoria per ogni gestore code di dati replicati del ripristino di emergenza (DR RDQM) è assegnata come due volumi logici separati per gestore code da questo gruppo di volumi. (Ogni gestore code richiede due volumi logici per supportare l'operazione di ripristino dell'istantanea, quindi ogni DR RDQM viene assegnato poco più del doppio della memoria specificata quando viene creato.) Per prestazioni ottimali, questo gruppo di volumi deve essere costituito da uno o più volumi fisici che corrispondono alle unità disco interne (preferibilmente SSD).
- Dopo aver creato il gruppo di volumi `drbdpool`, non eseguire altre operazioni. IBM MQ gestisce i volumi logici creati in `drbdpool` come e dove sono montati.

- Ogni nodo richiede un'interfaccia utilizzata per la replica dei dati. Deve avere una larghezza di banda sufficiente per supportare i requisiti di replica dato il carico di lavoro previsto di tutti i gestori code di dati replicati.

Per la massima tolleranza di errore, questa interfaccia deve essere una NIC (Network Interface Card) indipendente.

- DRBD richiede che ogni nodo utilizzato per RDQM abbia un nome host internet valido (il valore restituito da `uname -n`), come definito da RFC 952 modificato da RFC 1123.
- Se esiste un firewall tra i nodi utilizzati per DR RDQM, il firewall deve consentire il traffico tra i nodi sulle porte utilizzate per la replica. Viene fornito uno script di esempio, `/opt/mqm/samp/rdqm/firewalld/configure.sh`, che apre le porte necessarie se si sta eseguendo il firewall standard in RHEL. È necessario eseguire lo script come `root`. Se si utilizza un altro firewall, esaminare le definizioni di servizio `/usr/lib/firewalld/services/rdqm*` per vedere quali porte devono essere aperte. Lo script aggiunge le seguenti regole di servizio firewallD permanenti per DRBD e IBM MQ (è possibile modificare lo script per omettere le porte Pacemaker se non si utilizza HA):
 - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-drbd.xml` consente le porte TCP 7000-7100.
 - `MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/rdqm-mq.xml` consente la porta TCP 1414 (è necessario modificare lo script se si richiede una porta diversa)
- Se il sistema utilizza SELinux in una modalità diversa da quella permissiva, è necessario eseguire il seguente comando:

```
semanage permissive -a drbd_t
```

Requisiti di rete

Si consiglia di individuare i nodi utilizzati per il ripristino di emergenza in diversi data center.

È necessario essere consapevoli delle seguenti limitazioni:

- Le prestazioni peggiorano rapidamente con l'aumento della latenza tra i data center. IBM supporterà una latenza fino a 5 ms per la replica sincrona e 100 ms per la replica asincrona.
- I dati inviati attraverso il link di replicazione non sono soggetti ad alcuna ulteriore crittografia oltre a quella che potrebbe essere in atto utilizzando IBM MQ AMS.
- La configurazione di un gestore code RDQM per il ripristino di emergenza comporta un sovraccarico dovuto al requisito di replicare i dati tra i due nodi RDQM. La replica sincrona comporta un sovraccarico maggiore rispetto alla replica asincrona. Quando viene utilizzata la replica sincrona, le operazioni I/O del disco vengono bloccate finché i dati non vengono scritti su entrambi i nodi. Quando viene utilizzata la replica asincrona, i dati devono essere scritti solo nel nodo primario prima che l'elaborazione possa continuare.

Requisiti utente per l'utilizzo dei gestori code

Per creare, eliminare o configurare i gestori code di dati replicati (RDQM), è necessario essere l'utente `root` o disporre di un ID utente appartenente al gruppo `mqm` a cui viene concessa l'autorizzazione `sudo` per i seguenti comandi:

- `crtmqm`
- `dltmqm`
- `rdqmdr`

Un utente che appartiene al gruppo `mqm` può visualizzare lo stato di un DR RDQM utilizzando i seguenti comandi:

- `dspmq`
- `rdqmstatus`

L'utente mqm deve avere lo stesso UID su entrambi i server e il gruppo mqm deve avere lo stesso GID su entrambi i server.

Linux Creazione di un RDQM di ripristino di emergenza

Utilizzare il comando `crtmqm` per creare un gestore code di dati replicati (RDQM) che agisca come primario o secondario in una configurazione di ripristino di emergenza.

Informazioni su questa attività

È possibile creare un gestore code di dati replicati (RDQM) come utente nel gruppo mqm se l'utente può utilizzare sudo. Altrimenti, è necessario creare l'RDQM come root.

È necessario creare un gestore code RDQM DR primario su un nodo. È necessario quindi creare un'istanza secondaria dello stesso gestore code su un altro nodo. Le istanze primarie e secondarie hanno lo stesso nome e devono essere assegnate alla stessa quantità di memoria.

I seguenti punti forniscono alcune indicazioni sul dimensionamento del file system del gestore code:

1. Durante la creazione di un gestore code RDQM, viene allocato un file system per memorizzare i dati e i log del gestore code. È importante dimensionare questo file system in modo appropriato in modo che il gestore code possa registrare l'attività in corso nei propri log e memorizzare i messaggi dell'applicazione nelle code. Quando si ridimensiona il file system, considerare i requisiti di messaggistica di picco, la crescita futura del carico di lavoro e le interruzioni dell'applicazione che potrebbero causare l'accumulo di messaggi nelle code. Per istruzioni sul calcolo della dimensione del log di ripristino del gestore code, consultare [“Quanto deve essere grande il file system del log?” a pagina 675](#). Quando si calcolano i requisiti di memoria per i messaggi dell'applicazione, è necessario considerare la dimensione e il numero di messaggi, l'intestazione MQMD e le proprietà del messaggio di cui dispongono.
2. I file system del gestore code RDQM non possono essere ridimensionati dinamicamente. È necessario eseguire il backup e ripristinare un gestore code RDQM con un file system più grande, se richiesto, consultare [“Ridimensionamento del file system per un gestore code HA RDQM” a pagina 603](#).
3. È possibile limitare la dimensione delle singole code sul disco utilizzando attributi della coda locale, come MAXDEPTH e MAXFSIZE. Consultare [Modifica dei file della coda IBM MQ](#).
4. È necessario monitorare l'utilizzo del disco in corso e rispondere in modo appropriato se l'utilizzo del disco aumenta prima che l'utilizzo del file system diventi critico. L'utilizzo del file system può essere monitorato utilizzando le funzionalità della piattaforma / del sistema operativo o sottoscrivendo le metriche pubblicate negli argomenti di sistema IBM MQ descritti in [Metriche pubblicate negli argomenti di sistema](#).

Procedura

- Per creare un DR RDQM primario:
 - a) Immettere il seguente comando:

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -ip Port  
[other_crtmqm_options] [-fs size] QMname
```

dove:

-rr p

Specifica che si sta creando l'istanza primaria del gestore code.

-rt a | s

-rt s specifica che la configurazione DR utilizza la replica sincrona, **-rt a** specifica che la configurazione DR utilizza la replica asincrona. La replica asincrona è il valore predefinito.

-rl IP_locale

Specifica l'indirizzo IP locale da utilizzare per la replica DR di questo gestore code.

-ri IP_recupero

Specifica l'indirizzo IP dell'interfaccia utilizzata per la replica sul server che ospita l'istanza secondaria del gestore code.

-rn Nome_recupero

Specifica il nome del sistema che ospita l'istanza secondaria del gestore code. Il nome è il valore restituito se si esegue `uname -n` su tale server. È necessario creare esplicitamente un gestore code secondario su tale server.

-rp Porta

Specifica la porta da utilizzare per la replica DR.

altre opzioni `crtmqm_options`

Facoltativamente, è possibile specificare una o più delle seguenti opzioni **`crtmqm`** generali:

- -z
- -q
- -c *Testo*
- -d *DefaultTransmissionDefaultTransmission*
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa *utente|gruppo*
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -lc | -l
- -lla | -lln
- -lf *LogFileDimensione*
- -p *Porta*

-fs dimensione

Facoltativamente specifica la dimensione del file system da creare per il gestore code, ossia la dimensione del volume logico creato nel gruppo di volumi `drbdpool`. Viene creato anche un altro volume logico di tale dimensione, per supportare il ripristino dell'operazione di istantanea, in modo che la memoria totale per il DR RDQM sia poco più del doppio di quella specificata qui.

Dimensione è un valore numerico, specificato in GB. È possibile specificare un valore in MB immettendo il valore seguito dal carattere M. Ad esempio, per specificare una dimensione del file system di 3 GB, immettere 3. Per specificare una dimensione del filesystem di 1024 MB, immettere 1024M. (È anche possibile aggiungere un suffisso G allo stato esplicitamente GB.)

QMNAME

Specifica il nome del gestore code di dati replicato. Il nome è sensibile al maiuscolo/minuscolo.

Una volta completato il comando, viene emesso il comando che è necessario immettere sul nodo secondario per creare l'istanza secondaria del gestore code. È inoltre possibile utilizzare il comando **`rdqmdr`** sul nodo principale per richiamare il comando **`crtmqm`** che è necessario eseguire sul nodo secondario per creare il gestore code secondario, consultare [“Gestione delle caratteristiche primarie e secondarie di RDQM DR”](#) a pagina 631.

- Per creare un DR RDQM secondario:
 - a) Immettere il seguente comando sul nodo che deve ospitare le istanze secondarie di RDQM:

```
crtmqm -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

dove:

-rr s

Specifica che si sta creando l'istanza secondaria del gestore code.

-rt a | s

-rt s specifica che la configurazione DR utilizza la replica sincrona, **-rt a** specifica che la configurazione DR utilizza la replica asincrona.

-rl IP_locale

Specifica l'indirizzo IP locale da utilizzare per la replica DR di questo gestore code.

-ri IP_Primario

Specifica l'indirizzo IP dell'interfaccia utilizzata per la replica sul server che ospita l'istanza primaria del gestore code.

-rn Nome_Primario

Specifica il nome del sistema che ospita l'istanza primaria del gestore code. Il nome è il valore restituito se si esegue `uname -n` su tale server.

-rp Porta

Specifica la porta da utilizzare per la replica DR.

altre opzioni crtmqm_options

Facoltativamente, è possibile specificare una o più delle seguenti opzioni **crtmqm** generali:

– -z

-fs dimensione

Specifica la dimensione del filesystem da creare per il gestore code, ossia la dimensione del volume logico creato nel gruppo di volumi drbdpool. Se è stata specificata una dimensione non predefinita durante la creazione del gestore code primario, è necessario specificare lo stesso valore.

Dimensione è un valore numerico, specificato in GB. È possibile specificare un valore in MB immettendo il valore seguito dal carattere M. Ad esempio, per specificare una dimensione del file system di 3 GB, immettere 3. Per specificare una dimensione del filesystem di 1024 MB, immettere 1024M. (È anche possibile aggiungere un suffisso G allo stato esplicitamente GB.)

QMNAME

Specifica il nome del gestore code di dati replicato. Deve essere uguale al nome specificato per l'istanza primaria del gestore code. Notare che il nome è sensibile al maiuscolo / minuscolo.

Operazioni successive

Dopo aver creato le istanze primarie e secondarie del proprio gestore code, è necessario controllare lo stato su entrambi i nodi per verificare che entrambi siano corretti. Utilizzare il comando **rdqmstatus** su entrambi i nodi. I nodi devono visualizzare lo stato normale come descritto in “Visualizzazione dello stato DR RDQM” a pagina 632. Se non stanno visualizzando questo stato, eliminare l'istanza secondaria e ricrearla, facendo attenzione a utilizzare gli argomenti corretti.

Riferimenti correlati

[crtmqm](#)

Eliminazione di un RDQM DR

Utilizzare il comando **dltmqm** per eliminare un gestore code di dati replicati del ripristino di emergenza (RDQM).

Informazioni su questa attività

È necessario eseguire il comando per eliminare l'RDQM su entrambi i nodi RDQM primario e secondario. RDQM deve essere terminato per primo. È possibile eseguire il comando come un utente mqm se tale utente dispone dei privilegi sudo necessari. Altrimenti, è necessario eseguire il comando come root.

Procedura

- Per eliminare un DR RDQM, immettere il seguente comando:

Riferimenti correlati

[dltmqm](#)

MQ Adv. **Linux** *Migrazione di un gestore code per diventare un gestore code DR RDQM*

È possibile migrare un gestore code esistente per diventare un gestore code di dati replicati di DR (Disaster Recovery) (RDQM) eseguendo il backup dei relativi dati persistenti, quindi ripristinando i dati su un gestore code RDQM appena creato con lo stesso nome.

Informazioni su questa attività

I gestori code di dati replicati DR richiedono un volume logico dedicato (filesystem) e la configurazione della replica del disco. Questi componenti vengono configurati solo quando viene creato un nuovo gestore code. Un gestore code esistente può essere migrato per utilizzare RDQM eseguendo il backup dei dati persistenti, quindi ripristinando i dati su un gestore code RDQM appena creato con lo stesso nome. Questa procedura conserva la configurazione del gestore code, lo stato e i messaggi persistenti al momento della creazione del backup.

Nota: È possibile migrare un gestore code solo da una versione di IBM MQ uguale o inferiore alla versione in cui è installato RDQM. Anche il sistema operativo e l'architettura devono essere uguali. Altrimenti, è necessario creare un nuovo gestore code sulla piattaforma di destinazione; fare riferimento a [Spostamento di un gestore code su un sistema operativo differente](#).

È necessario soddisfare le seguenti condizioni prima di migrare un gestore code:

- Valutare i requisiti di ripristino di emergenza e consultare [“Ripristino di emergenza RDQM”](#) a pagina 617.
- Esaminare le applicazioni e i gestori code che si connettono al gestore code. Considerare le modifiche richieste per instradare le connessioni al nodo RDQM su cui è in esecuzione il gestore code.
- Eseguire il provisioning o identificare i nodi RDQM esistenti per la configurazione scelta. Per informazioni sui requisiti di sistema per RDQM, consultare [“Requisiti per la soluzione RDQM DR”](#) a pagina 620.
- Installare IBM MQ Advanced, che include la funzione RDQM, su ciascun nodo.
- Facoltativamente, verificare la configurazione RDQM utilizzando un gestore code di prova, che può quindi essere eliminato. La verifica della configurazione è consigliata per identificare e risolvere eventuali problemi prima di migrare il gestore code.
- Esaminare la configurazione della sicurezza per il gestore code, quindi replicare gli utenti e i gruppi locali richiesti su ciascun nodo RDQM.
- Esaminare il gestore code e la configurazione del canale per determinare se vengono utilizzate le uscite API, le uscite canale o le uscite di conversione dati. Installare le uscite richieste su ciascun nodo RDQM.
- Esaminare i servizi del gestore code che sono stati definiti, quindi installare e configurare i processi richiesti su ciascun nodo RDQM.

Procedura

1. Eseguire il backup del gestore code esistente:

- a) Arrestare il gestore code esistente emettendo un comando di attesa arresto `endmqm -wo` un comando di arresto immediato `endmqm -i`. Questa operazione è importante per garantire la congruenza dei dati nel backup.
- b) Determinare l'ubicazione della directory dei dati del gestore code visualizzando il file di configurazione IBM MQ, `mqs.ini`. Su Linux, questo file si trova nella directory `/var/mqm`. Per ulteriori informazioni su `mqs.ini`, consultare [“File di configurazione IBM MQ, mqs.ini”](#) a pagina 94.

Individuare la sezione `QueueManager` per il gestore code nel file. Se la stanza contiene una chiave denominata `DataPath`, il suo valore è la directory dei dati del gestore code. Se la chiave non

esiste, la directory dei dati del gestore code può essere determinata utilizzando i valori delle chiavi `Prefix` e `Directory`. La directory dei dati del gestore code è una concatenazione di questi valori, nel formato `prefisso/qmgrs/directory`. Per ulteriori informazioni sulla stanza `QueueManager`, consultare [“Stanza QueueManager del file mqs.ini” a pagina 105](#).

- c) Creare un backup della directory dei dati del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando `tar`. Ad esempio, per eseguire il back up della directory di dati per un gestore code è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo punto (punto):

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- d) Determinare l'ubicazione della directory di log del gestore code visualizzando IBM MQ file di configurazione del gestore code `qm.ini`. Questo file si trova nella directory dei dati del gestore code. Per ulteriori informazioni sul file, consultare [“File di configurazione del gestore code, qm.ini” a pagina 107](#).

La directory di log del gestore code è definita come il valore della chiave `LogPath` nella stanza `Log`. Per informazioni sulla sezione, consultare [“Stanza di log del file qm.ini” a pagina 141](#).

- e) Creare un backup della directory di log del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando `tar`. Ad esempio, per eseguire il back up della directory di log per un gestore code, è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo punto (punto):

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- f) Creare un backup dei repository dei certificati utilizzati dal gestore code se non si trovano nella directory dei dati del gestore code. Verificare che sia il file del database delle chiavi che il file `stash` delle password siano sottoposti a backup. Per informazioni sul repository delle chiavi del gestore code, consultare [Il repository delle chiavi SSL/TLS e Individuazione del repository delle chiavi per un gestore code](#). Per informazioni sull'individuazione dell'archivio chiavi AMS se il gestore code è configurato per utilizzare l'intercettazione MCA (Message Channel Agent) AMS, consultare [Intercettazione MCA \(Message Channel Agent\)](#).
- g) Il gestore code esistente non è più richiesto, quindi può essere eliminato. Tuttavia, laddove possibile, è necessario eliminare il gestore code esistente solo dopo che è stato ripristinato correttamente nel sistema di destinazione. Il differimento dell'eliminazione garantisce che il gestore code possa essere riavviato se il processo di migrazione non viene completato correttamente.

Nota: Se si rinvia l'eliminazione del gestore code esistente, non riavviarla. È importante che il gestore code rimanga chiuso perché ulteriori modifiche alla relativa configurazione o stato vengono perse durante la migrazione.

2. Preparare il nodo RDQM primario:

- a) Creare un nuovo gestore code RDQM con lo stesso nome del gestore code di cui è stato eseguito il backup. Assicurarsi che il file system assegnato per il gestore code RDQM da `crtmqm` sia abbastanza grande da contenere i dati, i log primari e i log secondari per il gestore code esistente, oltre a spazio aggiuntivo per l'espansione futura. Per informazioni su come creare un gestore code RDQM, consultare [“Creazione di un RDQM di ripristino di emergenza” a pagina 622](#).
- b) Determinare il nodo RDQM primario per il gestore code. Per informazioni su come determinare il nodo primario, vedere [rdqmqstatus \(visualizzazione dello stato RDQM\)](#).
- c) Sul nodo RDQM primario, se il gestore code RDQM è avviato, arrestarlo utilizzando il comando `endmqm -w` o `endmqm -i`.
- d) Determinare l'ubicazione delle directory di dati e di log per il gestore code RDQM (utilizzare i metodi descritti nei passi 1b e 1d).
- e) Eliminare il contenuto delle directory di log e dei dati del gestore code RDQM, ma non le directory stesse.

3. Ripristinare il gestore code sul nodo RDQM primario:

- a) Copiare i backup dei dati del gestore code e le directory di log nel nodo RDQM primario, oltre a eventuali backup separati dei repository dei certificati utilizzati dal gestore code.
- b) Ripristinare il backup della directory di dati del gestore code nella directory di dati vuota per il nuovo gestore code RDQM, assicurando che la proprietà e le autorizzazioni del file siano conservate. Se il backup è stato creato utilizzando il comando tar di esempio nel passaggio 1c , l'utente root può utilizzare il seguente comando per ripristinarlo:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- c) Ripristinare il backup della directory di log del gestore code nella directory di log vuota per il nuovo gestore code RDQM, verificando che le autorizzazioni e la proprietà del file siano conservate. Se il backup è stato creato utilizzando il comando tar di esempio nel passo 1e , l'utente root può utilizzare il seguente comando per ripristinarlo:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- d) Modificare il file di configurazione del gestore code ripristinato, `qm.ini`, nella directory dei dati per il gestore code RDQM. Aggiornare il valore della chiave `LogPath` nella stanza `Log` per specificare la directory di log per il gestore code RDQM.

Esaminare gli altri percorsi file definiti nel file di configurazione e aggiornarli, se necessario. Ad esempio, potrebbe essere necessario aggiornare i seguenti percorsi:

- Il percorso per i file di log degli errori generati dai servizi di messaggi diagnostici.
- Il percorso per le uscite richieste dal gestore code.
- Il percorso per i file di caricamento switch se il gestore code è un coordinatore della transazione XA.

- e) Se il gestore code è configurato per utilizzare l'intercettazione AMS Message Channel Agent (MCA), copiare il keystore AMS nella nuova installazione RDQM, quindi esaminare e aggiornare la configurazione. Il keystore deve essere disponibile su ciascun nodo RDQM, quindi se non è ubicato nel filesystem replicato per il gestore code, deve essere copiato su ciascun nodo. Per ulteriori informazioni, consultare [Intercettazione MCA \(Message Channel Agent\)](#).

- f) Verificare che il Gestore code sia visualizzato dal comando `dspmq` e che il suo stato sia riportato come terminato. Il seguente esempio mostra l'output di esempio per un gestore code DR RDQM:

```
$ dspmq -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DRRROLE(Primary)
```

- g) Verificare che i dati del gestore code ripristinati siano stati replicati sui nodi RDQM secondari utilizzando il comando `rdqmstatus` per visualizzare lo stato per il gestore code. Lo stato DR deve essere riportato come `Normal` su ciascun nodo. Il seguente esempio mostra l'output di esempio per un gestore code DR RDQM:

```
$ rdqmstatus -m QM1
Queue manager status:           Ended normally
Queue manager file system:      51MB used, 1.0GB allocated [5%]
DR role:                        Primary
DR status:                      Normal
DR type:                        Synchronous
DR port:                        3000
DR local IP address:           192.168.20.1
DR remote IP address:          192.168.20.2
```

- h) Avviare il gestore code sul nodo RDQM primario.
- i) Connettersi al gestore code e aggiornare il valore dell'attributo del gestore code `SSLKEYR` per specificare la nuova ubicazione del repository dei certificati del gestore code. Per impostazione predefinita, il valore di questo attributo è impostato su `queue_manager_data_directory/ssl/key`. Il repository certificati deve essere ubicato nella stessa ubicazione su ciascun nodo RDQM. Se il repository non si trova nel file system replicato per il gestore code, deve essere copiato su ciascun nodo.

- j) Esaminare le definizioni degli oggetti IBM MQ per il gestore code e aggiornare il valore degli attributi degli oggetti che fanno riferimento alle impostazioni di rete modificate, la directory di installazione di IBM MQ o la directory dei dati del gestore code, inclusi i seguenti oggetti:
- Indirizzi IP locali utilizzati dai listener (attributoIPADDR).
 - Indirizzi IP locali utilizzati dai canali (attributoLOCLADDR).
 - Indirizzi IP locali definiti per i canali riceventi del cluster (attributoCONNAME).
 - Gli indirizzi IP locali definiti per gli oggetti delle informazioni di comunicazione (attributoGRPADDR).
 - Percorsi di sistema definiti per le definizioni di processo e oggetto servizio.
- k) Arrestare e riavviare il gestore code per rendere effettive le modifiche.
- l) Ripetere il passo 3j per gestori code remoti, più le impostazioni equivalenti per le applicazioni, che si connettono al gestore code migrato, incluso:
- Nomi connessione canale (attributoCONNAME).
 - Le regole di autenticazione del canale che limitano le connessioni in entrata dal gestore code in base al relativo nome host o indirizzo IP.
 - Tabelle CCDT (Client Channel Definition Table), DNS (Domain Name Settings), instradamento di rete o informazioni di connessione equivalenti.
- m) Eseguire un failover gestito del gestore code su ciascun nodo RDQM per assicurarsi che la configurazione richiesta sia stata stabilita correttamente, fare riferimento a [“Passaggio a un nodo di recupero”](#) a pagina 637.

Ridimensionamento del filesystem per un gestore code DR RDQM

Per ridimensionare il filesystem per un gestore code di dati replicati (RDQM) di DR (Disaster Recovery) esistente, eseguire il backup dei propri dati persistenti, quindi ripristinare i dati su un gestore code RDQM appena creato con lo stesso nome ma con un filesystem di dimensione diversa.

Informazioni su questa attività

I gestori code di dati replicati DR richiedono un volume logico dedicato (filesystem) e la configurazione della replica del disco. Questi componenti vengono configurati solo quando viene creato un nuovo gestore code. Il file system non può essere ridimensionato dopo che è stato creato perché deve avere la stessa dimensione su ogni nodo. Per ridimensionare il filesystem per un gestore code di dati replicati esistente (RDQM), è possibile eseguire il backup dei dati persistenti, quindi ripristinare i dati su un gestore code RDQM appena creato che ha lo stesso nome ma un filesystem di dimensione diversa. Questa procedura conserva la configurazione del gestore code, lo stato e i messaggi persistenti al momento della creazione del backup.

Procedura

1. Eseguire il backup del gestore code RDQM esistente sul nodo RDQM primario:
 - a) Determinare il nodo RDQM primario per il gestore code. Per informazioni su come determinare il nodo primario, vedere [rdqmstatus \(visualizzazione dello stato RDQM\)](#).
 - b) Sul nodo RDQM primario, se il gestore code RDQM è avviato, arrestarlo utilizzando il comando **endmqm -w** o **endmqm -i**.
 - c) Determinare l'ubicazione della directory dei dati del gestore code visualizzando il file di configurazione IBM MQ, `mqs.ini`. Su Linux, questo file si trova nella directory `/var/mqm`. Per ulteriori informazioni su `mqs.ini`, consultare [“File di configurazione IBM MQ, mqs.ini”](#) a pagina 94.

Individuare la sezione `QueueManager` per il gestore code nel file. La directory dei dati del gestore code è il valore della chiave denominata `DataPath`. Per ulteriori informazioni sulla stanza `QueueManager`, consultare [“Stanza QueueManager del file mqs.ini”](#) a pagina 105.

- d) Creare un backup della directory dei dati del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando **tar**. Ad esempio, per eseguire il back up della directory di dati per un gestore code è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo carattere punto (.):

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- e) Determinare l'ubicazione della directory di log del gestore code visualizzando IBM MQ file di configurazione del gestore code `qm.ini`. Questo file si trova nella directory dei dati del gestore code. Per ulteriori informazioni sul file, consultare [“File di configurazione del gestore code, qm.ini” a pagina 107](#).

La directory di log del gestore code è definita come il valore della chiave `LogPath` nella stanza `Log`. Per informazioni sulla sezione, consultare [“Stanza di log del file qm.ini” a pagina 141](#).

- f) Creare un backup della directory di log del gestore code. Su Linux, è possibile eseguire questa operazione utilizzando il comando **tar**. Ad esempio, per eseguire il back up della directory di log per un gestore code, è possibile utilizzare il seguente comando. Notare l'ultimo parametro del comando, che è un singolo carattere punto (.):

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- g) Eliminare il gestore code RDQM esistente.

2. Ripristinare il gestore code con un filesystem della dimensione richiesta:

- a) Creare un nuovo gestore code RDQM con lo stesso nome del gestore code di cui è stato eseguito il backup. Assicurarsi che il file system assegnato per il gestore code RDQM da **crtmqm** sia la dimensione richiesta e che sia sufficientemente grande per contenere i dati, i log primari e i log secondari per il gestore code esistente, più dello spazio aggiuntivo per l'espansione futura. Per informazioni su come creare un gestore code RDQM, consultare [“Creazione di un RDQM di ripristino di emergenza” a pagina 622](#).
- b) Determinare il nodo RDQM primario per il gestore code. Per informazioni su come determinare il nodo primario, vedere [rdqmstatus \(visualizzazione dello stato RDQM\)](#).
- c) Sul nodo RDQM primario, se il gestore code RDQM è avviato, arrestarlo utilizzando il comando **endmqm -w** o **endmqm -i**.
- d) Sul nodo RDQM primario, stabilire la nuova ubicazione delle directory di dati e di log per il gestore code RDQM (utilizzare i metodi descritti ai passi 1c e 1e).
- e) Sul nodo RDQM primario, eliminare il contenuto dei dati del gestore code RDQM e le directory di log, ma non le directory stesse.
- f) Sul nodo RDQM primario, ripristinare il backup della directory di dati del gestore code nella directory di dati vuota per il nuovo gestore code RDQM, verificando che la proprietà e le autorizzazioni del file siano conservate. Se il backup è stato creato utilizzando il comando **tar** di esempio nel passo 1d, il seguente comando può essere utilizzato dall'utente root per ripristinarlo:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- g) Sul nodo RDQM principale, ripristinare il backup della directory di log del gestore code nella directory di log vuota per il nuovo gestore code RDQM, verificando che la proprietà del file e le autorizzazioni siano conservate. Se il backup è stato creato utilizzando il comando **tar** di esempio nel passo 1f, il seguente comando può essere utilizzato dall'utente root per ripristinarlo:

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- h) Sul nodo RDQM primario, modificare il file di configurazione del gestore code ripristinato, `qm.ini`, nella directory dei dati per il nuovo gestore code RDQM. Aggiornare il valore della chiave `LogPath` nella stanza `Log` per specificare la directory di log per il nuovo gestore code RDQM determinato al passo 2d. Esaminare gli altri percorsi file definiti nel file di configurazione e aggiornarli, se necessario. Ad esempio, potrebbe essere necessario aggiornare i seguenti percorsi:

- Il percorso per i file di log degli errori generati dai servizi di messaggi diagnostici.

- Il percorso per le uscite richieste dal gestore code.
 - Il percorso per i file di caricamento switch se il gestore code è un coordinatore della transazione XA.
- i) Verificare che il gestore code sia visualizzato dal comando **dspmqr** e che il relativo stato sia riportato come ended. Il seguente esempio mostra l'output di esempio per un gestore code DR RDQM:

```
$ dspmqr -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DR(Primary)
```

- j) Verificare che i dati del gestore code ripristinati siano stati replicati sul nodo RDQM secondario utilizzando il comando **rdqmstatus** per visualizzare lo stato del gestore code. Lo stato DR deve essere riportato come Normal su ciascun nodo. Il seguente esempio mostra l'output di esempio per un gestore code DR RDQM sul nodo primario:

```
$ rdqmstatus -m QM1
Queue manager status:      Running
CPU:                       0.00
Memory:                    123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Normal
DR type:                   Synchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:      192.168.20.2
```

Il seguente esempio mostra l'output di esempio per un gestore code DR RDQM sul nodo di recupero:

```
Queue manager status:      Ended immediately
DR role:                   Secondary
DR status:                 Normal
DR port:                   3000
DR local IP address:       192.168.20.2
DR remote IP address:      192.168.20.1
```

- k) Avviare il gestore code sul nodo RDQM primario.
- l) Eseguire una commutazione del gestore code sul nodo di ripristino per assicurarsi che la configurazione richiesta sia stata stabilita correttamente, consultare [“Passaggio a un nodo di recupero”](#) a pagina 637.

Archiviazione dello stato dell'applicazione persistente

È possibile memorizzare le informazioni sullo stato persistente relative alle applicazioni insieme ad altri dati del gestore code.

Ogni gestore code IBM MQ dispone di un filesystem dedicato per il relativo stato persistente, che include sia i dati della coda che il log di recupero. In una configurazione RDQM, il filesystem è supportato da un volume logico replicato tra i sistemi Linux (nodi). Il file system include una directory `userdata` che è possibile utilizzare per memorizzare le informazioni sullo stato persistente per le applicazioni. Quindi, quando un gestore code di dati replicati si sposta per essere eseguito su un altro nodo nella configurazione RDQM, è disponibile il contesto dell'applicazione e il contesto del gestore code. Vedere [Contenuto della directory su Unix e Linux Systems](#).

Se si sceglie di memorizzare lo stato dell'applicazione nella directory `userdata`, è necessario tenere presente che i dati scritti in questa ubicazione potrebbero consumare lo spazio su disco disponibile assegnato al gestore code. È necessario assicurarsi che sia disponibile spazio su disco sufficiente per il gestore code per scrivere i dati della coda, i log e altre informazioni sullo stato persistente.

La directory `userdata` ha la proprietà utente e gruppo `mqm` ed è leggibile in modo che gli utenti possano accedervi senza dover far parte del gruppo di amministratori IBM MQ (`mqm`). Non è possibile modificare le autorizzazioni della directory `userdata`, ma è possibile creare il contenuto in essa con qualsiasi proprietà e autorizzazione richiesti.

Durante un failover del gestore code RDQM, il gestore code viene terminato e il relativo file system viene smontato sul nodo RDQM corrente. Il filesystem viene quindi montato e il gestore code viene

riavviato su un altro nodo nella configurazione RDQM. Un file system non può essere smontato se un processo ha un handle aperto per uno dei suoi file. Per garantire che un failover del gestore code possa essere completato, se il filesystem del gestore code non può essere smontato, ai processi che hanno un handle di file aperto viene inviato un segnale SIGTERM, seguito da un SIGKILL se gli handle aperti non vengono rilasciati. Le applicazioni devono essere progettate per rispondere correttamente a SIGTERM. Se le applicazioni o i processi sono configurati come un servizio del gestore code, durante un failover gestito possono essere terminati durante l'arresto del gestore code prima che il filesystem venga smontato. Se un'applicazione o un processo non è configurato come servizio del gestore code o si verifica un failover non gestito, ad esempio una perdita del quorum, è probabile che i segnali vengano inviati per rilasciare il file system.

Linux *Gestione delle caratteristiche primarie e secondarie di RDQM DR*

È possibile modificare un gestore code di dati replicati di ripristino di emergenza secondario (DR RDQM) in un DR RDQM primario. È anche possibile modificare un'istanza primaria in un'istanza secondaria.

Informazioni su questa attività

Utilizzare il comando **rdqmdr** per modificare un'istanza secondaria di un RDQM nell'istanza primaria. Potrebbe essere necessario completare questa azione se si perde l'istanza primaria per qualche motivo. È quindi possibile avviare il gestore code e continuare l'esecuzione sul nodo di recupero.

Utilizzare il comando **rdqmdr** anche per modificare un'istanza primaria di un RDQM nell'istanza secondaria. Potrebbe essere necessario completare questa azione, ad esempio, se si stava riconfigurando il sistema.

È inoltre possibile utilizzare **rdqmdr** su un gestore code primario per richiamare il comando esatto necessario per creare un'istanza secondaria di tale gestore code sul nodo di recupero.

È possibile utilizzare il comando **rdqmdr** come utente nel gruppo mqm se l'utente può utilizzare sudo. Altrimenti, è necessario essere collegati come root.

Procedura

- Per modificare un'istanza secondaria di un DR RDQM in un'istanza primaria, immettere il comando seguente:

```
rdqmdr -m QMname -p
```

Questo comando non riesce se l'istanza primaria del gestore code è ancora in esecuzione e il collegamento di replica DR è ancora in funzione.

- Per modificare un'istanza principale del gestore code in un'istanza secondaria, immettere il comando seguente:

```
rdqmdr -m QMname -s
```

- Per visualizzare il comando **crtmqm** richiesto per configurare l'istanza secondaria di un gestore code, immettere il seguente comando sul nodo primario:

```
rdqmdr -d -m QMname
```

È possibile immettere il comando **crtmqm** restituito sul nodo secondario per creare l'istanza secondaria di RD RDQM.

Linux *Avvio, arresto e visualizzazione dello stato di un DR RDQM*

Utilizzare le varianti dei comandi di controllo IBM MQ standard per avviare, arrestare e visualizzare lo stato corrente di un gestore code di dati replicati del ripristino di emergenza (DR RDQM).

Informazioni su questa attività

È necessario eseguire i comandi che avviano, arrestano e visualizzano lo stato corrente di un gestore code di dati replicati (RDQM) come utente appartenente al gruppo mqm .

È necessario eseguire i comandi per avviare e arrestare un gestore code sul nodo primario per tale gestore code (ossia, il nodo su cui il gestore code è attualmente in esecuzione).

Procedura

- Per avviare un DR RDQM, immettere il seguente comando sul nodo primario di RDQM:

```
strmqm qmname
```

dove *qmname* è il nome dell'RDQM che si desidera avviare.

- Per arrestare un RDQM, immettere il seguente comando sul nodo primario del RDQM:

```
endmqm qmname
```

dove *qmname* è il nome dell'RDQM che si desidera arrestare.

- Per visualizzare lo stato di un RDQM, immettere il seguente comando:

```
dspmq -m QMname
```

Le informazioni sullo stato che vengono emesse dipendono dal fatto che si esegua il comando sul nodo primario o secondario di RDQM. Se eseguito sul nodo primario, viene visualizzato uno dei normali messaggi di stato restituiti da **dspmq** . Se si esegue il comando su un nodo secondario, viene visualizzato lo stato `Ended immediately` . Ad esempio, se **dspmq** viene eseguito sul nodo RDQM7, potrebbero essere restituite le seguenti informazioni:

```
QMNAME(DRQM8)          STATUS(Ended immediately)
QMNAME(DRQM7)          STATUS(Running)
```

È possibile utilizzare gli argomenti con **dspmq** per stabilire se un RDQM è configurato per il ripristino di emergenza e se è attualmente l'istanza primaria o secondaria:

```
dspmq -m QMname -o (dr | DR)
```

Viene visualizzata una delle seguenti risposte:

DRROLE()

Indica che il gestore code non è configurato per il ripristino di emergenza.

DRROLE(Primary)

Indica che il gestore code è configurato come DR primario.

DRROLE(Secondary)

Indica che il gestore code è configurato come secondario DR.

Riferimenti correlati

[dspmq](#)

[endmqm](#)

[strmqm](#)

Visualizzazione dello stato DR RDQM

È possibile visualizzare lo stato di tutti i gestori code di dati replicati di ripristino di emergenza (RDQM DR) su un nodo o informazioni dettagliate per un RDQM DR specificato.

Informazioni su questa attività

Utilizzare il comando **rdqmstatus** per visualizzare lo stato di tutti i RDQM DR o dei singoli RDQM.

Lo stato di riepilogo per un nodo visualizza anche le informazioni sul modulo kernel DRBD su cui si basa RDQM. Quando si aggiorna RDQM, è importante assicurarsi che la versione corretta del modulo kernel DRBD sia installata per la versione del kernel RHEL in esecuzione sul sistema. Lo stato visualizza la versione del kernel del sistema operativo, la versione del kernel per cui è stato creato il modulo DRBD, la versione DRBD e lo stato caricato del modulo del kernel DRBD.

È necessario essere un utente nel gruppo mqm per eseguire il comando **rdqmstatus**. È possibile eseguire il comando su uno dei nodi della coppia DR RDQM.

Procedura

- Per visualizzare lo stato di riepilogo di tutti gli RDQM DR su un nodo, eseguire il seguente comando su tale nodo:

```
rdqmstatus
```

Viene visualizzato lo stato degli RDQM DR sul nodo, ad esempio:

```
Node:                               mqhvm07.exampleco.com
OS kernel version:                  5.14.0-362.18.1
DRBD OS kernel version:              5.14.0-362.18.1
DRBD version:                        9.2.7
DRBD kernel module status:           Loaded

Queue manager name:                  DRQM8
Queue manager status:                 Ended immediately
DR role:                              Secondary

Queue manager name:                  DRQM7
Queue manager status:                 Running
DR role:                              Primary
```

Lo stato del modulo kernel DRBD è uno dei seguenti valori:

Caricato

Indica che il modulo DRBD è stato caricato.

Caricato parzialmente

Può verificarsi quando il modulo DRBD è stato caricato, ma non funziona correttamente a causa di una mancata corrispondenza.

Non caricato

Il modulo DRBD non è caricato. Può essere visualizzato su una configurazione appena installata, quando non è stato ancora creato alcun gestore code RDQM.

Non installato

Indica che il modulo DRBD non è installato. oppure che IBM MQ non è stato in grado di determinare la versione kernel del sistema operativo del modulo DRBD.

La versione precedentemente installata è ancora caricata

Questo stato può verificarsi se un nuovo modulo DRBD è installato mentre il modulo DRBD esistente è in esecuzione (ossia, un gestore code RDQM è in esecuzione). Il modulo appena installato viene riportato nello stato, ma non è il modulo attualmente in esecuzione.

- Per visualizzare lo stato di un particolare RDQM, immettere il seguente comando:

```
rdqmstatus -m qmname
```

La seguente tabella riepiloga le informazioni restituite.

Tabella 35. Attributi stato		
Attributo Stato	Valori possibili	Quando visualizzato
Stato gestore code	stato (come visualizzato da dspmq)	Sempre visualizzato

<i>Tabella 35. Attributi stato (Continua)</i>		
Attributo Stato	Valori possibili	Quando visualizzato
CPU	<i>n.nn%</i>	Visualizzato solo quando RDQM sul nodo corrente ha un ruolo primario
Memoria	<i>nnnMB</i>	Visualizzato solo quando RDQM sul nodo corrente ha un ruolo primario
File system del gestore code	<i>nnnMB</i> utilizzati, <i>n.nGB</i> allocati [<i>n%</i>]	Visualizzato solo quando RDQM sul nodo corrente ha un ruolo primario
Ruolo DR	Primario Secondario Sconosciuto	Sempre visualizzato
Stato DR	Normale	Funzionamento normale
	Sincronizzazione in corso	La sincronizzazione è in corso
	Partizionato	Il gestore code è stato avviato su entrambi i nodi mentre la rete di replica DR non è disponibile
	Sistema remoto non disponibile	La connessione all'altro nodo è stata persa
	Non congruente	Era in corso una sincronizzazione, ma è stata interrotta
	Ripristino all'istantanea	L'utente ha scelto di ripristinare l'istantanea acquisita quando il gestore code è entrato nello stato Incongruente.
	Sistema remoto non configurato	L'istanza primaria di RDQM è stata configurata, ma non è stata configurata alcuna istanza secondaria
	Negoziiazione non riuscita	Uno dei nodi è stato impostato sulla replica sincrona e l'altro sulla replica asincrona
Tipo DR	Sincrono o asincrono	Sempre visualizzato
Porta DR	<i>numero_porta</i> (la porta TCP/IP utilizzata per replicare i dati per questo gestore code)	Sempre visualizzato
Indirizzo IP locale DR	L'indirizzo IP locale da cui questo gestore code esegue la replica per DR	Sempre visualizzato
Indirizzo IP remoto DR	L'indirizzo IP remoto su cui questo gestore code sta eseguendo la replica per DR	Sempre visualizzato

Tabella 35. Attributi stato (Continua)		
Attributo Stato	Valori possibili	Quando visualizzato
Dati DR non sincronizzati	nKB	Visualizzato quando il nodo remoto non è disponibile o non è congruente
Avanzamento della sincronizzazione DR	n%	Visualizzato quando la sincronizzazione è in corso
Tempo stimato DR per il completamento	AAAA-MM-GG HH:MM:SS	Visualizzato quando la sincronizzazione è in corso
Avanzamento ripristino istantanea	n%	Visualizzato quando lo stato DR è Reverting to snapshot. Lo stato viene contato in basso, quindi 0% mostra il completamento
Ultimo DR sincronizzato	AAAA-MM-GG HH:MM:SS	Visualizzato quando i dati DR non sono sincronizzati (dopo la sincronizzazione iniziale). Fornisce l'ora e la data in cui i dati sono stati sincronizzati l'ultima volta.

Esempio

Esempio di stato normale sul nodo primario:

```
Queue manager status:      Running
CPU:                      0.00
Memory:                   123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                  Primary
DR status:                Normal
DR type:                  Synchronous
DR port:                  3000
DR local IP address:      192.168.20.1
DR remote IP address:     192.168.20.2
```

Esempio di stato normale su un nodo secondario:

```
Queue manager status:      Ended immediately
DR role:                   Secondary
DR status:                 Normal
DR port:                   3000
DR local IP address:       192.168.20.2
DR remote IP address:     192.168.20.1
```

Esempio di stato sul nodo primario quando è in corso la sincronizzazione:

```
Queue manager status:      Running
CPU:                      0.53
Memory:                   124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                  Primary
DR status:                Synchronization in progress
DR type:                  Synchronous
DR port:                  3000
DR local IP address:      192.168.20.1
DR remote IP address:     192.168.20.2
DR synchronization progress: 11.0%
DR estimated time to completion: 2017-09-06 14:55:05
```

Esempio di un nodo primario, che mostra che è partizionato:

```

Queue manager status:      Running
CPU:                       0.02
Memory:                    124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Partitioned
DR type:                   Synchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:     192.168.20.2

```

Esempio di un nodo primario, che mostra che non è sincronizzato con il nodo secondario:

```

Queue manager status:      Running
CPU:                       0.00
Memory:                    123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Remote unavailable
DR type:                   Asynchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:     192.168.20.2
DR out of sync data:       15932KB
DR last in sync:          2020-07-27 16:01:47

```

Esempio di uno stato di riepilogo che mostra una mancata corrispondenza tra la versione kernel del sistema operativo (RHEL 9.3) e il modulo kernel DRBD (destinato a RHEL 9.2). Anche se lo stato indica che il modulo kernel DRBD è caricato e il gestore code previsto è in esecuzione, in questa situazione è necessario aggiornare il modulo kernel DRBD con la versione di destinazione per il kernel del sistema operativo in esecuzione.

```

Node:                       mqhavam07.exampleco.com
OS kernel version:          5.14.0-362.18.1
DRBD OS kernel version:    5.14.0-284.11.1
DRBD version:               9.2.7+ptf.14
DRBD kernel module status:  Loaded

Queue manager name:         DRQM8
Queue manager status:      Ended immediately
DR role:                   Secondary

Queue manager name:         DRQM7
Queue manager status:      Running
DR role:                   Primary

```

Esempio di stato di riepilogo che mostra una mancata corrispondenza tra la versione del kernel del sistema operativo (RHEL 8.10) e il modulo del kernel DRBD (destinato a RHEL 8.8). In questo esempio la mancata corrispondenza della versione è più grave e il caricamento del modulo del kernel DRBD non riesce. QM3 è un gestore code DR e deve essere l'istanza primaria, ma poiché il modulo kernel DRBD non è stato completamente caricato, viene riportato come secondario con uno stato DR Unknown. Per risolvere questo errore, il modulo kernel DRBD deve essere aggiornato con la destinazione della versione per il kernel del sistema operativo in esecuzione.

```

Node:                       mqhavam57.exampleco.com
OS kernel version:          4.18.0-553
DRBD OS kernel version:    4.18.0-477
DRBD version:               9.2.7+ptf.14
DRBD kernel module status:  Partially loaded

Queue manager name:         QM3
Queue manager status:      Status not available
DR role:                   Secondary
DR status:                 Unknown

```

Riferimenti correlati

 [rdqmstatus](#)

Esistono diverse situazioni in cui è possibile passare al gestore code secondario in una configurazione di ripristino di emergenza.

Ripristino di emergenza

Dopo la perdita completa del gestore code primario sul sito principale, si avvia il gestore code secondario sul sito di recupero. Le applicazioni si riconnettono al gestore code sul sito di recupero e il gestore code secondario elabora i messaggi dell'applicazione. I passi intrapresi per ripristinare la configurazione precedente dipendono dalla causa dell'errore. Ad esempio, perdita completa del nodo principale rispetto a perdita temporanea.

Per le operazioni da eseguire in seguito ad una perdita temporanea del sito principale, consultare [“Passaggio a un nodo di recupero” a pagina 637](#). Per le operazioni da eseguire dopo un errore permanente, consultare [“Sostituzione di un nodo malfunzionante in una configurazione di ripristino di emergenza” a pagina 639](#).

Supporto test di ripristino di emergenza

È possibile verificare la configurazione del ripristino di emergenza passando temporaneamente all'istanza secondaria e controllando che le applicazioni possano connettersi correttamente. Si segue la stessa procedura di quando si passa da un errore temporaneo del nodo primario a un altro, consultare [“Passaggio a un nodo di recupero” a pagina 637](#).

Ripristino all'istantanea

Se si verifica un errore nel nodo primario mentre è in corso una sincronizzazione, è possibile ripristinare l'istantanea acquisita dei dati del gestore code secondario appena prima dell'inizio della sincronizzazione. Il secondario viene quindi ripristinato ad uno stato congruente e può essere eseguito come primario. Per ripristinare l'istantanea, rendere il secondario come principale, come descritto in [“Passaggio a un nodo di recupero” a pagina 637](#). È necessario controllare che il ripristino dell'istantanea sia stato completato (utilizzando il comando `rdqmstatus`) prima di avviare il gestore code.

Se si verifica una situazione di emergenza nel sito principale, eseguire le operazioni per passare al sito di ripristino.

Informazioni su questa attività

In seguito alla perdita del gestore code primario sul sito principale, si rende il gestore code secondario sul sito di ripristino nel sito primario e lo si avvia. Le applicazioni si riconnettono al gestore code sul sito di recupero e il gestore code elabora i messaggi dell'applicazione. È possibile utilizzare questa procedura anche per verificare il nodo di recupero.

Importante: È necessario assicurarsi che l'istanza primaria di un gestore code non sia in grado di essere eseguita o arrestata e resa un'istanza secondaria, prima di promuovere l'istanza secondaria originale. Altrimenti i dati partizionati possono accumularsi.

È necessario essere collegati come root o come utente che appartiene al gruppo mqm e che dispone della configurazione sudo necessaria.

Procedura

1. Se si utilizza questa procedura per verificare il gestore code secondario (ovvero, l'istanza primaria è ancora in esecuzione), è necessario arrestare l'istanza primaria e designarla nuovamente come istanza secondaria:

```
endmqm qmname
rdqmdr -m qmname -s
```

2. Rendere il gestore code secondario primario immettendo il seguente comando sul nodo di recupero:

```
rdqmdr -m qmname -p
```

3. Avviare il gestore code mediante il comando:

```
strmqm qmname
```

4. Verificare che le proprie applicazioni si riconnettano al gestore code sul gestore code di ripristino. Se sono stati definiti i canali con un elenco di nomi di connessione alternativi, specificando i gestori code primari e secondari, le applicazioni si conatteranno automaticamente al nuovo gestore code primario.

Operazioni successive

Quando il nodo in errore viene ripristinato, a condizione che il link tra i due nodi sia funzionante, il gestore code non è in grado di avviarlo su questo nodo perché è in esecuzione sul nodo di ripristino in cui è stata promossa l'istanza del gestore code secondario. Per tornare al normale funzionamento, è necessario arrestare il gestore code sul nodo di recupero, quindi promuovere nuovamente il gestore code sul nodo originale al ruolo primario.

Riferimenti correlati

[strmqm](#)

[rdqmdr](#)

Verifica del gestore code RDQM di ripristino

È possibile verificare che l'istanza di ripristino di un gestore code in una configurazione di ripristino di emergenza RDQM funzioni correttamente senza interrompere il sito principale.

Informazioni su questa attività

Si verifica il gestore code di ripristino disabilitando l'interfaccia tra i nodi principale e di ripristino. Si rende il gestore code secondario primario e si può quindi verificare il gestore code autonomo. Una volta completata la verifica, ripristinare l'interfaccia ed eliminare il gestore code di prova. Si ricrea quindi il gestore code come gestore code secondario nella configurazione del ripristino di emergenza.

Procedura

1. Disabilitare la connessione di rete tra il nodo principale e il nodo di recupero.
2. Sul nodo di recupero, rendere il gestore code il principale:

```
rdqmdr -m QMname -p
```

Dove *QMname* è il nome del gestore code.

3. Avviare il gestore code:

```
strmqm QMname
```

4. Connetti le applicazioni al gestore code e verifica che funzionino come previsto.
5. Terminare il gestore code:

```
endmqm QMname
```

6. Eliminare il gestore code:

```
dltmqm QMname
```

7. Ripristinare la connessione di rete tra le appliance principale e di ripristino.
8. Sul nodo principale, eseguire il seguente comando per richiamare il comando **crtmqm** utilizzato quando è stato configurato per la prima volta il ripristino di emergenza.

```
rdqmdr -d -m QMname
```

9. Eseguire il comando **crtmqm** risultante sul nodo di recupero per creare nuovamente un gestore code secondario. Il gestore code primario sul nodo principale sincronizza i propri dati con il gestore code secondario per aggiornarli.

Linux

Sostituzione di un nodo malfunzionante in una configurazione di ripristino di emergenza

Se si perde uno dei nodi in una configurazione di ripristino di emergenza, è possibile sostituire il nodo e ripristinare la configurazione di ripristino di emergenza seguendo questa procedura.

Informazioni su questa attività

Se si verifica una situazione di emergenza tale che il nodo nel sito principale è irreparabile, è possibile sostituire il nodo in errore mentre il gestore code viene eseguito sul nodo di recupero e quindi ripristinare la configurazione di ripristino di emergenza originale. Il nodo di sostituzione deve assumere l'identità del nodo non riuscito: il nome e l'indirizzo IP devono essere uguali.

È necessario essere collegati come root o come utente che appartiene al gruppo mqm e che dispone della configurazione sudo necessaria.

Procedura

A seguito della perdita del gestore code sul sito principale, effettuare le seguenti operazioni:

1. Sul nodo di recupero, eseguire i seguenti comandi per fare in modo che il gestore code secondario assuma il ruolo primario:

```
rdqmdr -m QMname -p
```

Dove *QMname* è il nome del gestore code.

2. Richiamare il comando che sarà necessario eseguire sul nodo primario di sostituzione per riconfigurare il ripristino di emergenza:

```
rdqmdr -m QMname -d
```

Copiare l'emissione di questo comando.

3. Eseguire il seguente comando per avviare il gestore code:

```
stmqm QMname
```

4. Assicurarsi che le applicazioni si riconnettano al gestore code sul nodo di recupero. Se sono stati definiti i canali con un elenco di nomi di connessione alternativi, specificando i gestori code primari e secondari, le applicazioni si conatteranno automaticamente al nuovo gestore code primario.
5. Sostituire il nodo in errore sul proprio sito principale e configurarlo in modo che abbia lo stesso nome e indirizzo IP utilizzato per il ripristino di emergenza sul nodo originale. Quindi, configurare il ripristino di emergenza eseguendo il comando **crtmqm** copiato nel passo 2. Ora si dispone di un'istanza secondaria del gestore code e l'istanza primaria sincronizza i suoi dati con l'istanza secondaria.
6. Terminare l'istanza primaria corrente.
7. Una volta completata la sincronizzazione, rendere nuovamente secondaria l'istanza primaria in esecuzione sul nodo di ripristino:

```
rdqmdr -m QMname -s
```

8. Sul nodo primario di sostituzione, rendere l'istanza secondaria del gestore code nell'istanza primaria:

```
rdqmdr -m QMname -p
```

9. Sul nodo primario di sostituzione, avviare il gestore code:

```
strmqm QMname
```

La configurazione è stata ripristinata come prima dell'errore nel sito principale.

Riferimenti correlati

[strmqm](#)

[rdqmdr](#)

[endmqm](#)

Risoluzione di un problema incongruente in DR RDQM

Uno stato DR di `inconsistent` può essere riportato se la sincronizzazione ha esito negativo tra le istanze primaria e secondaria di un gestore code.

Informazioni su questa attività

Uno stato incongruente viene riportato sull'istanza secondaria di un gestore code poiché la connessione di replica all'istanza primaria viene persa durante un'operazione di sincronizzazione. Potrebbe essere necessario intraprendere un'azione per risolvere questa situazione. Considerare la seguente sequenza di eventi:

1. Gestore code primario DR sincronizzato con il gestore code secondario DR
2. Link di replica perso tra primario e secondario
3. Link di replica ripristinato tra primario e secondario
4. Una risincronizzazione si verifica quando il gestore code secondario DR raggiunge il gestore code primario DR. Durante questo tempo, lo stato DR di `synchronization in progress` viene notificato per entrambi i gestori code.
5. Se la replica viene nuovamente persa durante la risincronizzazione, lo stato sul DR secondario viene riportato come `Inconsistent`.

Se il nodo che ospita il gestore code primario è ancora operativo e il link di replica può essere ripristinato, la risincronizzazione avviene automaticamente. Lo stato incongruente viene risolto senza intraprendere alcuna azione.

Se il nodo che ospita il gestore code primario non è più operativo, è possibile risolvere lo stato incongruente implementando un ripristino all'istantanea sul gestore code secondario. Questa operazione riporta i dati all'ultimo stato valido noto.

Procedura

Per risolvere uno stato incongruente:

1. Sul nodo di recupero, rendere l'istanza secondaria nell'istanza primaria:

```
rdqmdr -m qmname -p
```

Viene avviata l'operazione di ripristino dell'istantanea.

2. Sul nodo di recupero, controllare lo stato del gestore code per verificare quando l'operazione di ripristino dell'istantanea è completa:

```
rdqmstatus -m qmname
```

3. Quando lo stato del gestore code è `Normal`, avviare il gestore code:

```
strmqm qmname
```

Risoluzione di un problema partizionato (split brain) in DR RDQM

Un problema partizionato può verificarsi se entrambi i gestori code in una coppia di ripristino di emergenza vengono eseguiti contemporaneamente nel ruolo primario.

Informazioni su questa attività

Se è stata promossa l'istanza secondaria di un gestore code sul nodo di recupero mentre l'istanza primaria originale continuava ad essere eseguita sul nodo principale, si dispone effettivamente di due versioni dello stesso gestore code in esecuzione, ciascuna con la propria vista dei dati del gestore code. Lo stato DR per il gestore code su ciascun nodo è riportato come `Partitioned`.

È necessario decidere quale dei due gestori code ha la vista più corretta dei dati e conservare tale serie eliminando l'altro. Utilizzare il comando `rdqmdr` per completare questa operazione.

Ci sono due procedure. La prima descrive la conservazione dei dati dal nodo principale, la seconda descrive la conservazione dei dati dal nodo di ripristino.

Procedura

- Per conservare i dati dal gestore code sul nodo principale:
 - a) Verificare che entrambe le istanze del gestore code siano arrestate.
 - b) Specificare che il gestore code sul nodo di ripristino è il secondario:

```
rdqmdr -m qmname -s
```

- c) Specificare che il gestore code sul nodo principale è il principale:

```
rdqmdr -m qmname -p
```

Inizia la sincronizzazione, con i dati dal gestore code sul nodo principale copiati sul nodo di recupero.

- d) Verificare lo stato della sincronizzazione:

```
rdqmstatus -m qmname
```

- e) Una volta completata la sincronizzazione, avviare il gestore code sul nodo principale:

```
stimqm qmname
```

- Per conservare i dati dal Gestore code sul nodo di recupero:
 - a) Verificare che entrambe le istanze del gestore code siano arrestate.
 - b) Specificare che il gestore code sul nodo principale è il secondario:

```
rdqmdr -m qmname -s
```

- c) Specificare che il gestore code sul nodo di ripristino è il principale:

```
rdqmdr -m qmname -p
```

La sincronizzazione inizia con i dati del gestore code sul nodo di recupero copiati sul nodo principale.

- d) Verificare lo stato della sincronizzazione:

```
rdqmstatus -m qmname
```

- e) Una volta completata la sincronizzazione, degradare il gestore code sul nodo di recupero:

```
rdqmdr -m qmname -s
```

- f) Promuovere il gestore code sul nodo principale e avviarlo:

```
rdqmdr -m qmname -p  
stimqm qmname
```

Modifica degli indirizzi IP nelle configurazioni del ripristino di emergenza

Se si modificano gli indirizzi IP di una delle interfacce in una configurazione di ripristino di emergenza, la replica non è più possibile tra i due nodi.

Se è necessario modificare gli indirizzi IP per l'interfaccia di replica per uno dei nodi DR, è necessario utilizzare la seguente procedura:

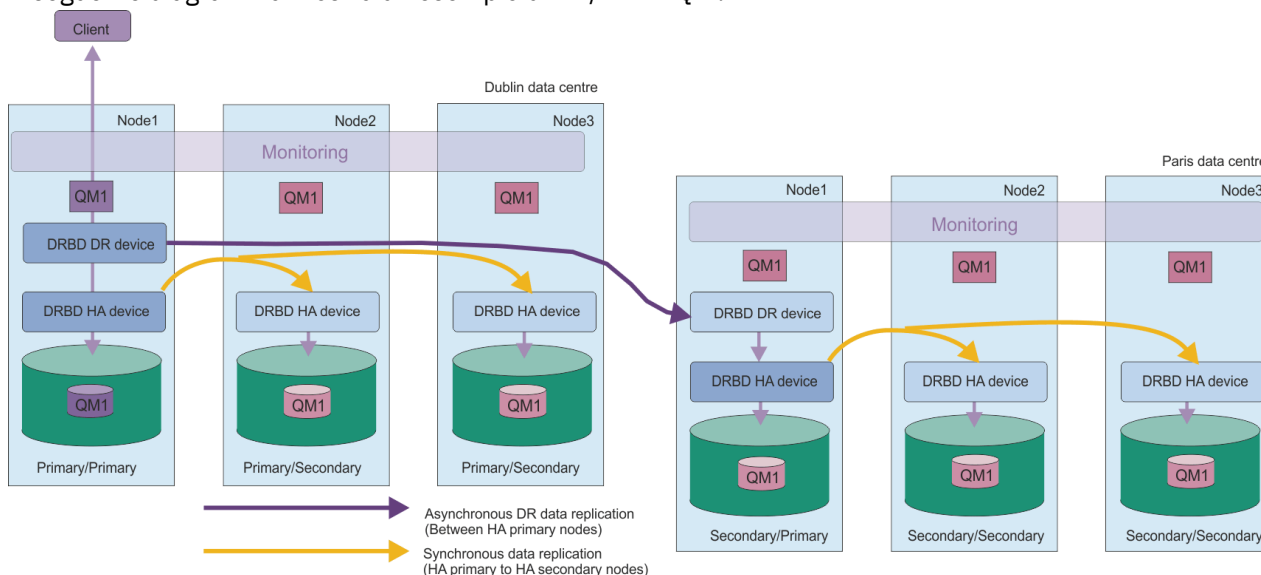
1. Sul nodo primario, eseguire il backup dei gestori code DR ed eliminarli. Sul nodo di recupero, eliminare i gestori code. Consultare [“Backup e ripristino dei dati del gestore code IBM MQ”](#) a pagina 696 e [“Eliminazione di un RDQM DR”](#) a pagina 624.
2. Ricreare i gestori code DR, specificando i nuovi indirizzi IP e ripristinare i backup, consultare [“Creazione di un RDQM di ripristino di emergenza”](#) a pagina 622 e [“Backup e ripristino dei dati del gestore code IBM MQ”](#) a pagina 696.

Linux RDQM disaster recovery e alta disponibilità

È possibile configurare un gestore code di dati replicati (RDQM) che viene eseguito su un gruppo ad alta disponibilità su un sito, ma può eseguire il failover su un altro gruppo ad alta disponibilità su un altro sito se si verifica una situazione di emergenza che rende il primo gruppo non disponibile. Questo è noto come DR/HA RDQM.

Un RDQM DR/HA combina le funzioni di un RDQM ad elevata disponibilità (consultare [“Alta disponibilità RDQM”](#) a pagina 588) e un RDQM di ripristino di emergenza (consultare [“Ripristino di emergenza RDQM”](#) a pagina 617).

Il seguente diagramma mostra un esempio di DR/HA RDQM.



La replica tra RDQM DR/HA sul sito principale e il sito di ripristino di emergenza è sempre asincrona. Con la replica asincrona, operazioni come IBM MQ PUT o GET completano e ritornano all'applicazione prima che l'evento venga replicato sul gestore code secondario.

È possibile avere due siti attivi invece di siti 'main' e 'recovery', se necessario, quindi alcuni dei RDQM DR/HA vengono eseguiti su un sito e alcuni sull'altro durante il normale funzionamento. Se si verifica un'emergenza e un sito diventa non disponibile, tutti gli RDQM DR/HA vengono eseguiti sullo stesso gruppo HA nello stesso sito.

Ogni gruppo HA è configurato nello stesso modo di un gruppo HA ordinario. È possibile definire indirizzi IP mobili per un RDQM DR/HA in ogni gruppo HA. L'indirizzo IP mobile può essere lo stesso o diverso per ogni gruppo HA.

Non è possibile aggiornare un RDQM esistente in modo che sia DR/HA RDQM, è necessario creare un RDQM DR/HA. (Se necessario, è possibile eseguire il backup dei dati di un RDQM esistente, eliminarli,

ricrearli come RDQM DR/HA e quindi ripristinare i dati, consultare [“Backup e ripristino dei dati del gestore code IBM MQ”](#) a pagina 696.)

Per configurare RDQM DR/HA, è necessario completare le seguenti operazioni principali:

1. Configurare un gruppo HA sul sito 'principale'.
2. Configurare un gruppo HA sul sito 'recovery'.
3. Creare un RDQM primario / primario DR/HA su un nodo del gruppo HA nel sito 'principale'.
4. Creare RDQM DR/HA primari / secondari sugli altri due nodi nel sito 'principale'.
5. Definire un indirizzo IP mobile per un'applicazione per accedere al DR/HA RDQM quando è in esecuzione su uno dei nodi del gruppo HA sul sito 'principale'.
6. Creare un RDQM DR/HA secondario / primario su un nodo del gruppo HA sul sito 'recovery'.
7. Creare RDQM secondario / secondario DR/HA sugli altri due nodi nel sito 'recovery'.
8. Definire un indirizzo IP mobile per un'applicazione per accedere al DR/HA RDQM quando è in esecuzione su uno qualsiasi dei nodi del gruppo HA sul sito 'recovery'.

I dettagli su ciascuno di questi passi sono forniti nei seguenti argomenti.

Linux *Requisiti per una soluzione DR/HA RDQM*

I requisiti per la soluzione DR/HA RDQM sono gli stessi della soluzione HA RDQM e della soluzione DR RDQM.

Per i requisiti per le parti HA della configurazione, consultare [“Requisiti per la soluzione RDQM HA”](#) a pagina 590.

Per i dettagli della parte DR della configurazione, vedere [“Requisiti per la soluzione RDQM DR”](#) a pagina 620.

Linux *Configurazione di gruppi HA per RDQM DR/HA*

È necessario creare un gruppo HA sia sul sito principale che su quello di ripristino. Se si dispone di un gruppo HA esistente su entrambi i siti, è possibile creare RDQM DR/HA in tale gruppo HA. (Gli RDQM esistenti continueranno a funzionare come prima.)

La procedura è la stessa descritta per l'alta disponibilità RDQM, consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593.

Quando si definisce un gruppo ad alta disponibilità, specificare gli indirizzi IP utilizzati per il controllo e la replica da ogni nodo nel file `rdqm.ini`. Quando si crea un gruppo HA per supportare RDQM DR/HA, è possibile anche specificare gli indirizzi IP utilizzati per la replica DR dal gruppo HA che si sta definendo e gli indirizzi IP utilizzati per la replica DR dai nodi nell'altro gruppo HA della coppia DR. (Se non si specificano gli indirizzi IP di replica DR nel file `rdqm.ini`, è possibile specificarli nella riga comandi quando si crea un RDQM DR/HA.)

Se si sta configurando un gruppo HA esistente, è possibile aggiungere gli indirizzi IP di replica DR al file `rdqm.ini` esistente. Non è necessario eseguire nuovamente `rdqmadm` dopo l'aggiornamento `rdqm.ini`, ma è necessario aggiornare `rdqm.ini` prima di creare RDQM DR/HA.

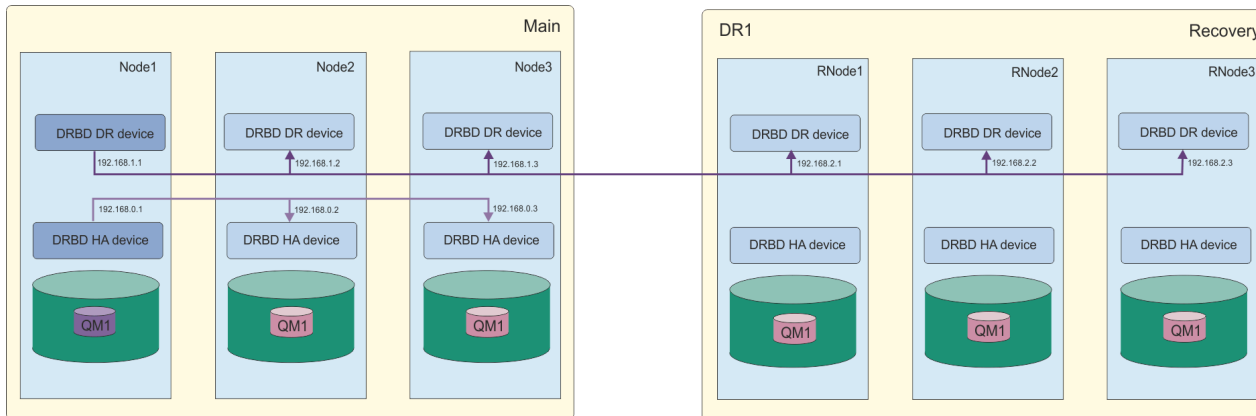
Utilizzare l'attributo `DR_Replication` nelle stanze Node per specificare le interfacce di replica DR sul gruppo HA che si sta definendo, ad esempio:

```
Node:
  Name=Node1
  HA_Replication=192.168.0.1
  DR_Replication=192.168.1.1
Node:
  Name=Node2
  HA_Replication=192.168.0.2
  DR_Replication=192.168.1.2
Node:
  Name=Node3
  HA_Replication=192.168.0.3
  DR_Replication=192.168.1.3
```

Utilizzare la sezione DRGroup per specificare gli indirizzi di replica DR del gruppo HA remoto, ad esempio:

```
DRGroup:
  Name=DR1
  DR_Replication=192.168.2.1
  DR_Replication=192.168.2.2
  DR_Replication=192.168.2.3
```

Il seguente diagramma illustra questa configurazione:



Se non si specificano gli indirizzi IP di replica DR per i nodi nel gruppo HA locale nel file `rdqm.ini` o sulla riga comandi quando si crea un RDQM DR/HA, le interfacce `HA_Replication` definite per ciascun nodo vengono utilizzate per la replica DR. È necessario specificare gli indirizzi di replica DR del gruppo HA remoto nel file `rdqm.ini` o sulla riga comandi `crtmqm`.

Linux Creazione di RDQM DR/HA

Utilizzare il comando `crtmqm` per creare un gestore code di dati replicati (RDQM) in una configurazione DR/HA.

Informazioni su questa attività

È possibile creare un RDQM DR/HA come utente nel gruppo `mqm` se l'utente può utilizzare `sudo`. Altrimenti, è necessario creare l'RDQM come `root`.

È necessario creare un numero di RDQM DR/HA:

- Sul gruppo HA sul sito 'principale':
 - Sul nodo in cui si desidera che il gestore code venga eseguito in condizioni normali, creare il DR/HA RDQM primario / primario.
 - Su ciascuno degli altri due nodi nel gruppo HA, creare un RDQM DR/HA primario / secondario.
- Sul gruppo HA sul sito 'recovery':
 - Sul nodo in cui verrà eseguito il gestore code se si verifica un failover sul sito di recupero, creare il DR/HA RDQM secondario / primario. È possibile utilizzare l'output del comando quando è stato creato il gestore code primario / primario sul sito 'principale'.
 - Su ognuno degli altri due nodi nel gruppo HA, creare un RDQM secondario / secondario DR/HA.

A tutte le istanze del gestore code deve essere assegnato lo stesso nome e la stessa quantità di memoria.

I seguenti punti forniscono alcune indicazioni sul dimensionamento del file system del gestore code:

1. Durante la creazione di un gestore code RDQM, viene allocato un file system per memorizzare i dati e i log del gestore code. È importante dimensionare questo file system in modo appropriato in modo che il gestore code possa registrare l'attività in corso nei propri log e memorizzare i messaggi dell'applicazione nelle code. Quando si ridimensiona il file system, considerare i requisiti di messaggistica di picco, la crescita futura del carico di lavoro e le interruzioni dell'applicazione che potrebbero causare l'accumulo di messaggi nelle code. Per istruzioni sul calcolo della dimensione del

log di ripristino del gestore code, consultare [“Quanto deve essere grande il file system del log?”](#) a pagina 675. Quando si calcolano i requisiti di memoria per i messaggi dell'applicazione, è necessario considerare la dimensione e il numero di messaggi, l'intestazione MQMD e le proprietà del messaggio di cui dispongono.

2. I file system del gestore code RDQM non possono essere ridimensionati dinamicamente. È necessario eseguire il backup e ripristinare un gestore code RDQM con un file system più grande, se richiesto, consultare [“Ridimensionamento del file system per un gestore code HA RDQM”](#) a pagina 603.
3. È possibile limitare la dimensione delle singole code sul disco utilizzando attributi della coda locale, come MAXDEPTH e MAXFSIZE. Consultare [Modifica dei file della coda IBM MQ](#).
4. È necessario monitorare l'utilizzo del disco in corso e rispondere in modo appropriato se l'utilizzo del disco aumenta prima che l'utilizzo del file system diventi critico. L'utilizzo del file system può essere monitorato utilizzando le funzionalità della piattaforma / del sistema operativo o sottoscrivendo le metriche pubblicate negli argomenti di sistema IBM MQ descritti in [Metriche pubblicate negli argomenti di sistema](#).

Procedura

- Per creare il DR/HA RDQM primario / primario:
 - a) Immettere il seguente comando:

```
crtmqm -sx -rr p
      [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
      (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
      -rp DRPort
      [-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]
      [-g ApplicationGroup] [-oa user|group]
      [-t TrigInt] [-u DeadQ] [-x MaxUMsgs]
      [-lp LogPri] [-ls LogSec]
      [-lc | -ll | -lla | -lln] [-lf LogFileSize]
      [-p Port] [-fs FilesystemSize] QMgrName
```

dove:

-sx

Indica che il ruolo HA iniziale è primario.

-rr p

Indica che il ruolo DR iniziale è primario.

-rl DRLocalIP1, DRLocalIP2, DRLocalIP3

Facoltativamente, specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito locale (ovvero, il sito 'principale'). Se non specificato, vengono utilizzati gli indirizzi IP specificati nel file `rdqm.ini`.

-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3

Specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito remoto (ovvero, il sito 'recovery'). È necessario specificare questo parametro o il parametro `-rn`.

-rn GroupName

Specificare il nome del gruppo HA remoto come specificato nel file `rdqm.ini`. È necessario specificare `-ri` o `-rn`.

-rp Porta

Specifica la porta da utilizzare per la replica DR.

altre opzioni `crtmqm_options`

Facoltativamente, è possibile specificare una o più delle seguenti opzioni `crtmqm` generali:

- -z
- -q
- -c *Testo*
- -d *DefaultTransmissionDefaultTransmission*
- -h *MaxHandles*

- *-g ApplicationGroup*
- *-oa utente | gruppo*
- *-t TrigInt*
- *-u DeadQ*
- *-x MaxUMsgs*
- *-lp LogPri*
- *-ls LogSec*
- *-lc | -l*
- *-lla | -lln*
- *-lf LogFileDimensione*
- *-p Porta*

-fs dimensione

Facoltativamente, specifica la dimensione del file system da creare per il gestore code, ossia la dimensione del volume logico creato nel gruppo di volumi drbdpool. Viene creato anche un altro volume logico di tale dimensione, per supportare il ripristino dell'operazione di istantanea, in modo che la memoria totale per il DR RDQM sia poco più del doppio di quella specificata qui.

Dimensione è un valore numerico, specificato in GB. È possibile specificare un valore in MB immettendo il valore seguito dal carattere M. Ad esempio, per specificare una dimensione del file system di 3 GB, immettere 3. Per specificare una dimensione del filesystem di 1024 MB, immettere 1024M. (È anche possibile aggiungere un suffisso G allo stato esplicitamente GB.)

QMNAME

Specifica il nome del gestore code di dati replicato. Il nome è sensibile al maiuscolo/minuscolo.

Una volta completato il comando, viene emesso il comando che è possibile immettere sul sito di recupero per creare l'istanza secondaria / primaria del gestore code.

- Per creare un RDQM DR/HA primario / secondario sugli altri due nodi nel gruppo HA:
 - a) Immettere il comando seguente su ciascun nodo:

```

crtmqm -sxs -rr p
          [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
          (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
          -rp DRPort
          [-fs FilesystemSize] QMgrName

```

dove:

-sxs

Indica che il ruolo HA iniziale è secondario.

-rr p

Indica che il ruolo DR iniziale è primario.

-rl DRLocalIP1, DRLocalIP2, DRLocalIP3

Facoltativamente, specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito locale (ovvero, il sito 'principale'). Se non specificato, vengono utilizzati gli indirizzi IP specificati nel file `rdqm.ini`.

-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3

Specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito remoto (ovvero, il sito 'recovery'). È necessario specificare questo parametro o il parametro `-rn`.

-rn GroupName

Specificare il nome del gruppo HA remoto come specificato nel file `rdqm.ini`. È necessario specificare `-ri` o `-rn`.

-rp Porta

Specifica la porta da utilizzare per la replica DR.

-fs dimensione

Specifica la dimensione del file system da creare per il gestore code, ovvero la dimensione del volume logico creato nel gruppo di volumi drbdpool. Se è stata specificata una dimensione non predefinita quando si crea l'RDQM primario / primario, è necessario specificare qui lo stesso valore.

Dimensione è un valore numerico, specificato in GB. È possibile specificare un valore in MB immettendo il valore seguito dal carattere M. Ad esempio, per specificare una dimensione del file system di 3 GB, immettere 3. Per specificare una dimensione del filesystem di 1024 MB, immettere 1024M. (È anche possibile aggiungere un suffisso G allo stato esplicitamente GB.)

QMNAME

Specifica il nome dell'RDQM primario / secondario. Deve essere uguale al nome specificato per l'istanza primaria / primaria di RDQM. Notare che il nome è sensibile al maiuscolo / minuscolo.

- Per creare un RDQM DR/HA secondario / primario sul nodo su cui verrà eseguito il gestore code in caso di failover sul sito di ripristino:

- a) Utilizzare l'output del comando quando si è creato il DR/HA primario / primario sul sito principale oppure immettere il seguente comando:

```
critmqm -sx -rr s
      [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
      (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
      -rp DRPort
      [-fs FilesystemSize] QMgrName
```

-sx

Indica che il ruolo HA iniziale è primario.

-rr s

Indica che il ruolo DR iniziale è secondario.

-rl DRLocalIP1, DRLocalIP2, DRLocalIP3

Facoltativamente, specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito locale (ovvero, il sito 'recovery'). Se non specificato, vengono utilizzati gli indirizzi IP specificati nel file `rdqm.ini`.

-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3

Specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito remoto (ovvero, il sito 'principale'). È necessario specificare questo parametro o il parametro `-rn`.

-rn GroupName

Specificare il nome del gruppo HA remoto come specificato nel file `rdqm.ini`. È necessario specificare `-ri` o `-rn`.

-rp Porta

Specifica la porta da utilizzare per la replica DR.

-fs dimensione

Facoltativamente, specifica la dimensione del file system da creare per il gestore code, ossia la dimensione del volume logico creato nel gruppo di volumi drbdpool. Viene creato anche un altro volume logico di tale dimensione, per supportare il ripristino dell'operazione di istantanea, in modo che la memoria totale per il DR RDQM sia poco più del doppio di quella specificata qui.

QMNAME

Specifica il nome del gestore code di dati replicato. Il nome è sensibile al maiuscolo/minuscolo.

- Per creare un RDQM secondario / secondario HA/DR sugli altri due nodi sul sito di ripristino:

- a) Immettere il comando seguente su ciascun nodo:

```
critmqm -sxs -rr s
      [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
      (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
      -rp DRPort
      [-fs FilesystemSize] QMgrName
```

-sxs

Indica che il ruolo HA iniziale è primario.

-rr s

Indica che il ruolo DR iniziale è secondario.

-rl DRLocalIP1, DRLocalIP2, DRLocalIP3

Facoltativamente, specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito locale. Se non specificato, vengono utilizzati gli indirizzi IP specificati nel file `rdqm.ini`.

-ri DRRemoteIP1, DRRemoteIP2, DRRemoteIP3

Specificare gli indirizzi IP delle interfacce DR sui tre nodi sul sito remoto. È necessario specificare questo parametro o il parametro `-rn`.

-rn GroupName

Specificare il nome del gruppo HA remoto come specificato nel file `rdqm.ini`. È necessario specificare `-ri` o `-rn`.

-rp Porta

Specifica la porta da utilizzare per la replica DR.

-fs dimensione

Facoltativamente, specifica la dimensione del file system da creare per il gestore code, ossia la dimensione del volume logico creato nel gruppo di volumi `drbdpool`. Viene creato anche un altro volume logico di tale dimensione, per supportare il ripristino dell'operazione di istantanea, in modo che la memoria totale per il DR RDQM sia poco più del doppio di quella specificata qui.

QMNAME

Specifica il nome del gestore code di dati replicato. Il nome è sensibile al maiuscolo/minuscolo.

Nota: Quando si crea un RDQM, il successivo numero di porta libera superiore a 7000 viene assegnato per il collegamento di replica HA. Se viene rilevato che la porta scelta è utilizzata da un'altra applicazione, il comando `crtmqm` ha esito negativo con l'errore AMQ6543 e tale porta viene aggiunta a un elenco di esclusione. È possibile eliminare le istanze secondarie del gestore code, quindi eseguire nuovamente il comando `crtmqm`.

Operazioni successive

Una volta create tutte le RDQM DR/HA, è necessario controllare lo stato sulle istanze primaria / primaria e secondaria / primaria per verificare che siano tutte corrette. Utilizzare il comando `rdqmstatus` sui nodi. I nodi devono visualizzare lo stato normale come descritto in [“Visualizzazione dello stato del gruppo DR/HA RDQM e HA”](#) a pagina 650. Se non stanno visualizzando questo stato, eliminare l'istanza secondaria / primaria e ricrearla, facendo attenzione a utilizzare gli argomenti corretti.

Attività correlate

[“Creazione di RDQM DR/HA”](#) a pagina 644

Utilizzare il comando `crtmqm` per creare un gestore code di dati replicati (RDQM) in una configurazione DR/HA.

Riferimenti correlati

[crtmqm](#)

 [Eliminazione di un RDQM DR/HA](#)

Utilizzare il comando `dlmqm` per eliminare un gestore code di dati replicati DR/HA (RDQM).

Informazioni su questa attività

È necessario eseguire il comando per eliminare l'RDQM sia sul nodo primario / primario che sul nodo secondario / primario. RDQM deve essere terminato per primo. È possibile eseguire il comando come un utente `mqm` se tale utente dispone dei privilegi `sudo` necessari. Altrimenti, è necessario eseguire il comando come `root`.

Procedura

- Per cancellare un RDQM DR/HA, immettere il seguente comando:

```
dltmqm RDQM_name
```

Riferimenti correlati

[dltmqm](#)

Linux Creazione di un indirizzo IP mobile

È possibile creare indirizzi IP mobili per ciascuno dei gruppi HA in una configurazione RDQM DR/HA.

Un indirizzo IP mobile consente a un client di utilizzare lo stesso indirizzo IP per un RDQM DR/HA indipendentemente dal nodo in un gruppo HA su cui è in esecuzione. Se i tuoi due gruppi HA hanno reti private / isolate per la connettività dell'applicazione, è possibile definire lo stesso indirizzo IP mobile per entrambi i gruppi. Devi ancora definire tale indirizzo IP mobile due volte, tuttavia, una volta su ognuno dei gruppi HA.

Gli indirizzi IP mobili vengono creati ed eliminati utilizzando lo stesso metodo di un RDQM HA. Consultare [“Creazione ed eliminazione di un indirizzo IP mobile”](#) a pagina 606.

Linux Avvio, arresto e visualizzazione dello stato di un RDQM DR/HA

Utilizzare le varianti dei comandi di controllo IBM MQ standard per avviare, arrestare e visualizzare lo stato corrente di un RDQM DR/HA.

Informazioni su questa attività

È necessario eseguire i comandi che avviano, arrestano e visualizzano lo stato corrente di un RDQM DR/HA come un utente che appartiene ai gruppi mqm e haclient.

È necessario eseguire i comandi per avviare e arrestare un gestore code sul nodo primario per tale gestore code.

Procedura

- Per avviare un RDQM, immettere il seguente comando sul nodo primario di RDQM:

```
strmqm qmname
```

dove *qmname* è il nome del DR/HA RDQM che si desidera avviare.

RDQM viene avviato e Pacemaker avvia la gestione di RDQM. È necessario specificare l'opzione `-ns` con `strmqm` se si desidera specificare altre opzioni `strmqm`.

- Per arrestare un RDQM, immettere il seguente comando sul nodo primario del DR/HA RDQM:

```
endmqm qmname
```

dove *qmname* è il nome dell'RDQM che si desidera arrestare.

Pacemaker cessa di gestire RDQM, quindi l'RDQM viene terminato. Tutti gli altri parametri `endmqm` possono essere utilizzati quando si arresta un RDQM.

- Per visualizzare lo stato di un RDQM, immettere il seguente comando:

```
dspmqr -m QMname
```

Le informazioni sullo stato che vengono emesse dipendono dal fatto che si esegua il comando sul nodo primario o secondario di RDQM. Se eseguito sul nodo primario, viene visualizzato uno dei normali messaggi di stato restituiti da `dspmqr`. Se si esegue il comando su un nodo secondario, viene

visualizzato lo stato Ended `immediately`. Ad esempio, se **dspmq** viene eseguito sul nodo RDQM7, potrebbero essere restituite le seguenti informazioni:

```
QMNAME(DRQM8)          STATUS(Ended immediately)
QMNAME(DRQM7)          STATUS(Running)
```

È possibile utilizzare gli argomenti con `dspmq` per stabilire se un RDQM è configurato per il ripristino di emergenza e se è attualmente l'istanza primaria o secondaria:

```
dspmq -m QMname -o (dr | DR)
```

Viene visualizzata una delle seguenti risposte:

DRROLE()

Indica che il gestore code non è configurato per il ripristino di emergenza.

DRROLE(Primary)

Indica che il gestore code è configurato come DR primario.

DRROLE(Secondary)

Indica che il gestore code è configurato come secondario DR.

Utilizzare il comando **dspmq -o all** per visualizzare le informazioni sul ripristino di emergenza e sull'alta disponibilità per RDQM DR/HA. Ad esempio, se si esegue **dspmq -o all** sul nodo in cui è in esecuzione DR/HA RDQM, vengono visualizzate le seguenti informazioni sullo stato:

```
QMNAME(TESTQM1)          STATUS(Running) HA(Replicated)
DRROLE(Primary)
```

Riferimenti correlati

[dspmq \(visualizza gestori code\)](#)

[endmqm \(fine gestore code\)](#)

[strmqm \(avvio gestore code\)](#)

Azioni della risorsa non riuscite nelle configurazioni DR/HA

Le azioni delle risorse non riuscite si verificano quando il componente Pacemaker di una configurazione ad alta disponibilità RDQM rileva alcuni problemi con una risorsa su uno dei nodi in un gruppo HA.

Le azioni delle risorse non riuscite potrebbero verificarsi su una delle configurazioni HA in una configurazione RDQM DR/HA. È possibile utilizzare il comando **rdqmstatus** per visualizzare le azioni delle risorse non riuscite e il comando **rdqmclean** per cancellarle (una volta risolta la causa dell'errore). Il processo è lo stesso delle configurazioni HA RDQM senza il componente DR. Per ulteriori dettagli, vedere [“Azioni della risorsa non riuscite”](#) a pagina 609.

Attività correlate

[“Visualizzazione dello stato del gruppo DR/HA RDQM e HA”](#) a pagina 650

È possibile visualizzare lo stato HA e il ruolo DR dei gestori code di dati replicati DR/HA (RDQM).

[“Visualizzazione dello stato del gruppo RDQM e HA”](#) a pagina 610

È possibile visualizzare lo stato del gruppo HA e dei singoli gestori code di dati replicati (RDQM).

Riferimenti correlati

[rdqmclean](#)

[rdqmstatus](#)

 Linux

Visualizzazione dello stato del gruppo DR/HA RDQM e HA

È possibile visualizzare lo stato HA e il ruolo DR dei gestori code di dati replicati DR/HA (RDQM).

Informazioni su questa attività

Utilizzare il comando **rdqmstatus** per visualizzare lo stato di singoli RDQM o ottenere una panoramica dello stato di tutti gli RDQM noti al gruppo HA.

Lo stato di riepilogo per un nodo visualizza anche le informazioni sul modulo kernel DRBD su cui si basa RDQM. Quando si aggiorna RDQM, è importante assicurarsi che la versione corretta del modulo kernel DRBD sia installata per la versione del kernel RHEL in esecuzione sul sistema. Lo stato visualizza la versione del kernel del sistema operativo, la versione del kernel per cui è stato creato il modulo DRBD, la versione DRBD e lo stato caricato del modulo del kernel DRBD.

Nota: Si noti che, in una configurazione HA/DR, la configurazione DR utilizza sempre la replica asincrona, mentre la configurazione HA utilizza sempre la replica sincrona. Questi valori non vengono visualizzati nell'output del comando `rdqmstatus -m qmgr` in una configurazione HA/DR combinata.

È necessario essere un utente nei gruppi `mqm` e `haclient` per eseguire il comando `rdqmstatus`. È possibile eseguire il comando su uno qualsiasi dei nodi in uno dei gruppi HA.

Procedura

- Per visualizzare lo stato di riepilogo di un nodo e degli RDQM che fanno parte della configurazione HA:

```
rdqmstatus
```

Viene visualizzata l'identità del nodo su cui è stato eseguito il comando e lo stato degli RDQM nella configurazione HA, più il ruolo DR corrente, ad esempio:

```
Node:                               main-alice
OS kernel version:                  5.14.0-362.18.1
DRBD OS kernel version:             5.14.0-362.18.1
DRBD version:                       9.2.7
DRBD kernel module status:         Loaded

Queue manager name:                 RDQM1
Queue manager status:               Running elsewhere
HA current location:                main-charlie
HA preferred location:              main-charlie
HA blocked location:                None

Queue manager name:                 RDQM9
Queue manager status:               Running elsewhere
HA current location:                main-bob
HA preferred location:              main-bob
HA blocked location:                None
DR role:                            Primary

Queue manager name:                 RDQM7
Queue manager status:               Running
HA current location:                This node
HA preferred location:              This node
HA blocked location:                None
DR role:                            Primary
```

In questo esempio RDQM7 e RDQM8 sono entrambi RDQM DR/HA, mentre RDQM1 è un RDQM HA, che non è configurato per passare a un sito di ripristino di emergenza.

Lo stato del modulo kernel DRBD è uno dei seguenti valori:

Caricato

Indica che il modulo DRBD è stato caricato.

Caricato parzialmente

Può verificarsi quando il modulo DRBD è stato caricato, ma non funziona correttamente a causa di una mancata corrispondenza.

Non caricato

Il modulo DRBD non è caricato. Può essere visualizzato su una configurazione appena installata, quando non è stato ancora creato alcun gestore code RDQM.

Non installato

Indica che il modulo DRBD non è installato, oppure che IBM MQ non è stato in grado di determinare la versione kernel del sistema operativo del modulo DRBD.

La versione precedentemente installata è ancora caricata

Questo stato può verificarsi se un nuovo modulo DRBD è installato mentre il modulo DRBD esistente è in esecuzione (ossia, un gestore code RDQM è in esecuzione). Il modulo appena installato viene riportato nello stato, ma non è il modulo attualmente in esecuzione.

- Per visualizzare lo stato di un particolare gestore code su tutti i nodi nel gruppo HA, immettere il seguente comando:

```
rdqmstatus -m qmname
```

dove *qmname* è il nome dell'RDQM per cui si desidera visualizzare lo stato. Viene visualizzato lo stato di RDQM sul nodo corrente, seguito da un riepilogo dello stato degli altri due nodi dalla prospettiva del nodo corrente.

- Per visualizzare lo stato di un particolare gestore code su tutti i nodi nel gruppo HA, inclusi i dettagli di eventuali azioni di risorse non riuscite, immettere il seguente comando:

```
rdqmstatus -m qmname -a
```

dove *qmname* è il nome dell'RDQM per cui si desidera visualizzare lo stato. Viene visualizzato lo stato di RDQM sul nodo corrente, seguito da un riepilogo dello stato degli altri due nodi dalla prospettiva del nodo corrente. Questo è seguito dai dettagli relativi a tutte le azioni di risorsa non riuscite associate a RDQM.

La seguente tabella riepiloga le informazioni sul nodo corrente che possono essere restituite dal comando `rdqmstatus -m qmname` per un RDQM.

Attributo Stato	Valori possibili	Quando visualizzato
Nome nodo	<i>NODENAME</i>	Sempre visualizzato
Stato gestore code	stato del gestore code (uno degli stati validi per il comando dspmq)	Sempre visualizzato
CPU	<i>n.nn%</i>	Visualizzato solo quando RDQM è in esecuzione su questo nodo
Memoria	<i>nnn</i> MB utilizzati	Visualizzato solo quando RDQM è in esecuzione su questo nodo
File system del gestore code	<i>nnn</i> MB utilizzati, <i>y.y</i> GB assegnati [<i>z%</i>]	Visualizzato solo quando RDQM è in esecuzione su questo nodo
Ruolo HA	Primario Secondario Sconosciuto	Sempre visualizzato
Stato HA	Tutti i nodi in standby Questo nodo è in standby Nodi remoti in standby Misto	Tutti i nodi in standby Nodo corrente in standby Entrambi i nodi remoti in standby Stato differente per ogni nodo remoto
Controllo HA	Abilitato/a Disabilitato/a Sconosciuto	Sempre visualizzato. Mostra se RDQM è sotto il controllo Pacemaker

Tabella 36. Stato nodo corrente (Continua)

Attributo Stato	Valori possibili	Quando visualizzato
Ubicazione preferita HA	Nessuna Questo nodo Sconosciuto <i>NODENAME</i>	Sempre visualizzato
Ubicazione bloccata HA	Nessuno - L'esecuzione del gestore code non è bloccata su alcun nodo Questo nodo - Il gestore code è bloccato dall'esecuzione sul nodo corrente a causa di una o più azioni di risorsa non riuscite <i>nodename</i> - L'esecuzione del gestore code è bloccata su <i>nodename</i> a causa di una o più azioni di risorsa non riuscite <i>nodename1, nodename2</i> - L'esecuzione del gestore code è bloccata su <i>nodename1</i> e <i>nodename2</i> a causa di una o più azioni della risorsa non riuscite Tutti i nodi - L'esecuzione del gestore code è bloccata su tutti i nodi a causa di una o più azioni di risorse non riuscite	Sempre visualizzato
Interfaccia IP mobile HA	<i>nome_interfaccia</i>	Sempre visualizzato
Indirizzo IP mobile HA	<i>IPV4_address</i>	Sempre visualizzato
Ruolo DR	Primario Secondario Secondario in sospeso Sconosciuto	Sempre visualizzato

Tabella 36. Stato nodo corrente (Continua)

Attributo Stato	Valori possibili	Quando visualizzato
Stato DR	Normale Sincronizzazione in corso Partizionato Sistema remoto non disponibile Non congruente Ripristino all'istantanea Sistema remoto non configurato Negoziazione non riuscita	Va tutto bene. Sincronizzazione in corso. L'utente ha avviato la coda su ciascun nodo mentre il La rete di replica DR era non disponibile. La connessione all'altro nodo è stato perso. Era in corso una sincronizzazione ma è stata interrotta. L'utente ha scelto di tornare a l'istantanea che è stata acquisita quando il gestore code ha immesso il Stato incongruente. Il primario è stato configurato Ma il secondario non l'ha fatto. La negoziazione iniziale tra i nodi primario e secondario non è riuscita. Ciò può essere causato da tipi di replica incompatibili o se il nodo secondario è configurato con una dimensione del file system inferiore.
Stato DR (su nodo secondario HA)	Consultare <i>HA_Primary_Node</i>	Visualizzato sui nodi secondari HA come stato DR è noto solo sul nodo primario HA.
Porta DR	La porta TCP/IP utilizzata per replicare i dati per questo gestore code.	Sempre visualizzato.
Indirizzo IP locale DR	L'indirizzo IP locale che verrà utilizzato da questo gestore code per la replica DR	Sempre visualizzato.
Elenco indirizzi IP remoti DR	Gli indirizzi IP remoti che questo gestore code utilizzerà per la replica DR. Un elenco separato da virgole di tre indirizzi IP.	Sempre visualizzato.
Indirizzo IP remoto corrente DR	L'IP remoto corrente a cui è connesso questo gestore code per la replica DR.	Per un primario HA con una connessione DR attiva.
Indirizzo IP remoto corrente DR (su nodo secondario HA)	Consultare <i>HA_Primary_Node</i>	Visualizzato su un nodo secondario HA come la connessione DR è solo sul nodo primario HA
Dati DR non sincronizzati	xKB	Visualizzato quando il nodo remoto non è disponibile o è incongruente.
Avanzamento della sincronizzazione DR	s%	Visualizzato quando è in corso una sincronizzazione.

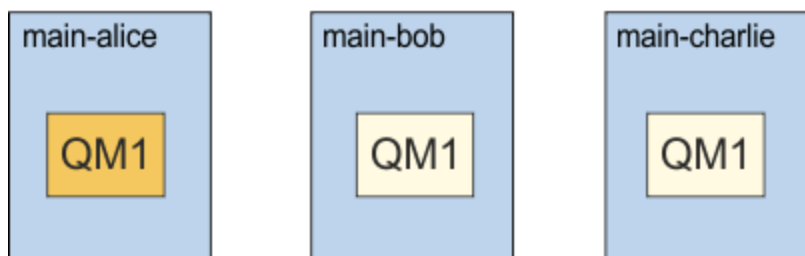
Tabella 36. Stato nodo corrente (Continua)

Attributo Stato	Valori possibili	Quando visualizzato
Tempo stimato DR per il completamento	aaaa-mm-gg hh:mm:ss	Visualizzato quando è in corso una sincronizzazione.
Avanzamento ripristino istantanea	s%	Visualizzato quando lo stato DR è "Ripristino istantanea"
Ultimo DR sincronizzato	aaaa-mm-gg hh:mm:ss	Visualizzato quando i dati DR non sono sincronizzati (dopo la sincronizzazione iniziale). Fornisce l'ora e la data in cui i dati sono stati sincronizzati l'ultima volta.

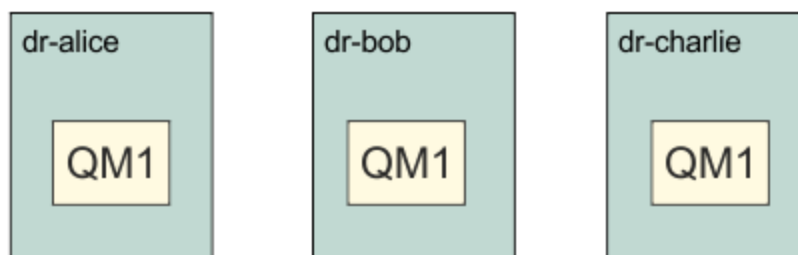
Esempio

Questi esempi illustrano il comando `rdqmstatus -m qm1` eseguito su vari nodi della seguente configurazione DR/HA:

main site



dr site



Esempio di stato normale su un nodo primario DR e primario HA:

```

Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
    
```

```

Node:                main-bob
HA status:           Normal

Node:                main-charlie
HA status:           Normal

```

Esempio di stato normale su un nodo che è il DR primario e un HA secondario:

```

Node:                main-bob
Queue manager status: Running elsewhere
HA role:             Secondary
HA status:           Normal
HA control:          Enabled
HA current location: main-alice
HA preferred location: main-alice
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role:             Primary
DR status:           See main-alice
DR port:             3000
DR local IP address: 192.168.1.2
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: See main-alice

Node:                main-alice
HA status:           Normal

Node:                main-charlie
HA status:           Normal

```

Esempio di stato normale su un nodo che è il secondario DR e un primario HA:

```

Node:                dr-alice
Queue manager status: Ended immediately
HA role:             Primary
HA status:           Normal
HA control:          Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role:             Secondary
DR status:           Normal
DR port:             3000
DR local IP address: 192.168.2.1
DR remote IP address list: 192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address: 192.168.1.1

Node:                dr-bob
HA status:           Normal

Node:                dr-charlie
HA status:           Normal

```

Esempio di stato normale su un nodo che è il secondario DR e un secondario HA:

```

Node:                dr-bob
Queue manager status: Ended immediately
HA role:             Secondary
HA status:           Normal
HA control:          Enabled
HA current location: dr-alice
HA preferred location: dr-alice
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role:             Secondary
DR status:           See dr-alice
DR port:             3000
DR local IP address: 192.168.2.2
DR remote IP address list: 192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address: See dr-alice

Node:                dr-alice
HA status:           Normal

```



```
Node: dr-charlie
HA status: Normal
```

Esempio di sincronizzazione DR in corso su un nodo che è un primario DR e un primario HA:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
DR synchronization progress: 11.0%
DR estimated time to completion: 2018-09-06 14:55:05

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal
```

Esempio di DR partizionato su un nodo che è un DR primario e HA primario:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Partitioned
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
DR out of sync data: 372KB

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal
```

Esempio di DR non sincronizzato su un nodo che è un DR primario e HA primario:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
```

```

DR role: Primary
DR status: Remote unavailable
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: Unknown
DR out of sync data: 372KB
DR last in sync: 2020-02-02 20:22:02

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal

```

Esempio di HA non sincronizzato su un nodo che è un DR primario e HA primario:

```

Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA blocked location: None
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1

Node: main-bob
HA status: Inconsistent
HA out of sync data: 15932KB
HA last in sync: 2020-02-02 20:22:02

Node: main-charlie
HA status: Normal

```

Esempio di uno stato di riepilogo che mostra una mancata corrispondenza tra la versione kernel del sistema operativo (RHEL 9.3) e il modulo kernel DRBD (destinato a RHEL 9.2). Anche se lo stato riporta che il modulo del kernel DRBD è caricato e il gestore code è in esecuzione, è necessario aggiornare il modulo del kernel DRBD con la versione destinata al kernel del sistema operativo in esecuzione in questa situazione.

```

Node: main-alice
OS kernel version: 5.14.0-362.18.1
DRBD OS kernel version: 5.14.0-284.11.1
DRBD version: 9.2.7+ptf.14
DRBD kernel module status: Loaded

Queue manager name: QM1
Queue manager status: Running
HA current location: This node
HA preferred location: This node
HA blocked location: None
DR role: Primary

```

Esempio di uno stato di riepilogo che mostra una mancata corrispondenza tra la versione kernel del SO (RHEL 9.3) e il modulo kernel DRBD (destinato a RHEL 9.0). In questo esempio la mancata corrispondenza della versione è più grave e il caricamento del modulo del kernel DRBD non riesce. QM1 è un gestore code HA/DR e si sposta su un altro nodo, il suo stato HA è sconosciuto e il suo stato DR è sconosciuto. Per risolvere questo errore, il modulo kernel DRBD deve essere aggiornato con la destinazione della versione per il kernel del sistema operativo in esecuzione.

```

Node: main-alice
OS kernel version: 5.14.0-362.18.1
DRBD OS kernel version: 5.14.0-70.13.1
DRBD version: 9.2.7+ptf.14

```

DRBD kernel module status:	Partially loaded
Queue manager name:	QM1
Queue manager status:	Running elsewhere
HA status:	Unknown
HA current location:	main-bob
HA preferred location:	This node
HA blocked location:	None
DR role:	Primary
DR status:	Unknown

Riferimenti correlati

Linux [rdqmstatus](#)

Linux *Funzionamento in un ambiente DR/HA*

Quando si opera in un ambiente DR/HA ci sono considerazioni separate per l'alta disponibilità e il ripristino di emergenza.

Se il nodo su cui è in esecuzione un RDQM DR/HA ha esito negativo, l'RDQM esegue automaticamente il failover su un altro nodo in tale gruppo HA. Se l'intero sito ha esito negativo, è necessario avviare manualmente RDQM sul nodo preferito nel gruppo HA sul sito di ripristino. Le considerazioni qui riportate sono le stesse di un normale RDQM DR, consultare [“Funzionamento in un ambiente di ripristino di emergenza”](#) a pagina 637 per ulteriori informazioni.

Se uno dei nodi ha esito negativo completamente e deve essere sostituito, consultare [“Sostituzione di un nodo malfunzionante in una configurazione di ripristino di emergenza”](#) a pagina 639 e [“Sostituzione di un nodo malfunzionante in una configurazione ad alta disponibilità”](#) a pagina 617 per istruzioni.

Linux *Sostituzione di un nodo non riuscito in una configurazione DR/HA*

Se uno dei nodi in uno dei tuoi gruppi HA ha esito negativo, puoi sostituirlo.

Informazioni su questa attività

La procedura varia a seconda se il nodo che si sta sostituendo è un nodo primario o un nodo secondario nella configurazione DR. In entrambi i casi, il nuovo nodo deve avere una configurazione identica al nodo che si sta sostituendo, ovvero, deve avere lo stesso nome host, gli stessi indirizzi IP e così via.

Si potrebbe anche incontrare la situazione in cui si è completamente perso il gruppo HA nel proprio sito principale o di ripristino e si deve sostituire l'intero gruppo HA.

Procedura

- Per un nodo di sostituzione primario nella configurazione DR, completare la seguente procedura sul nuovo nodo:
 - a) Creare un file `rdqm.ini` che corrisponda ai file sugli altri nodi, quindi eseguire il comando `rdqmadm -c` (consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593).
 - b) Eseguire il comando `crtmqm -sxs -rr p qmanager` per ricreare ogni DR/HA RDQM (consultare [“Creazione di RDQM DR/HA”](#) a pagina 644).
- Per un nodo di sostituzione secondario nella configurazione DR, completare la seguente procedura sul nuovo nodo:
 - a) Creare un file `rdqm.ini` che corrisponda ai file sugli altri nodi, quindi eseguire il comando `rdqmadm -c` (consultare [“Definizione del cluster Pacemaker \(gruppo HA\)”](#) a pagina 593).
 - b) Eseguire il comando `crtmqm -sx -rr s qmanager` per ricreare ciascun RDQM DR/HA (consultare [“Creazione di RDQM DR/HA”](#) a pagina 644).
- Per sostituire un intero gruppo HA, completare la seguente procedura:
 - a) Se si perde l'intero gruppo HA sul sito primario DR (ovvero, il sito principale), è necessario seguire la procedura per eseguire un failover gestito sul sito secondario DR per continuare ad eseguire RDQM DR/HA (consultare [“Funzionamento in un ambiente di ripristino di emergenza”](#) a pagina 637). (Se

si perde un intero gruppo HA nel sito di ripristino, i RDQM DR/HA continuano ad essere eseguiti nel sito principale senza l'intervento dell'utente.)

- b) Creare nuovamente il gruppo HA sui tre nodi di sostituzione, come descritto in [“Configurazione di gruppi HA per RDQM DR/HA”](#) a pagina 643.
- c) Ricreare le RDQM DR/HA sul nuovo gruppo HA come descritto in [“Creazione di RDQM DR/HA”](#) a pagina 644.
- d) Se necessario, eseguire un failover gestito dal sito di ripristino al sito principale.

Linux **Esempio di lavoro DR/HA RDQM**

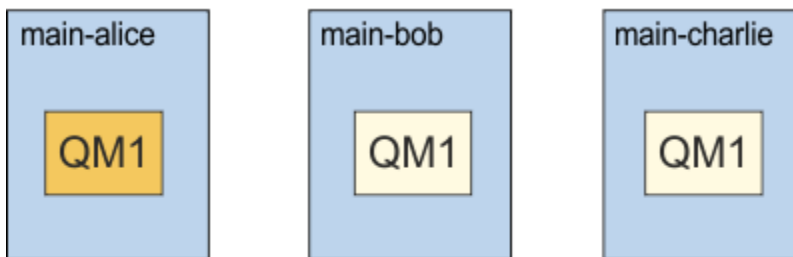
Questo esempio mostra come creare ed eliminare DR/HA RDQM.

Creazione di un RDQM DR/HA

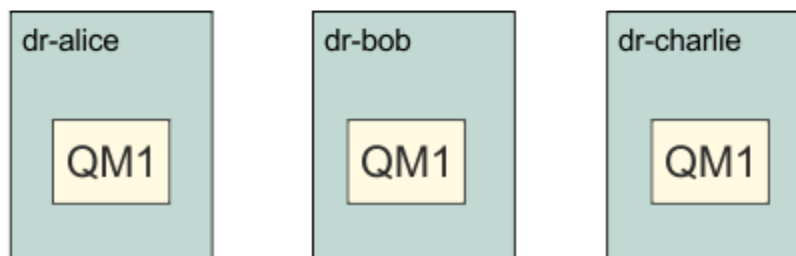
La configurazione di esempio ha due siti, denominati 'main' e 'dr'. Ogni sito ha tre nodi, denominati 'alice', 'bob' e 'charlie'. I nodi hanno un nome completo composto dal nome e dal nome del sito, quindi 'main-alice', 'dr-alice' e così via.

La seguente procedura crea un RDQM DR/HA denominato QM1 che viene eseguito su main - alice. Il nodo principale - alice è il primario HA e DR.

main site



dr site



Se gli indirizzi IP DR locali e remoti sono specificati nel file `rdqm.ini`, non è necessario specificare alcun indirizzo IP sulla riga comandi e un RDQM DR/HA denominato QM1 può essere creato eseguendo il seguente comando su main - alice:

```
crtmqm -sx -rr p -rn DR1 -rp 7001 QM1
```

Se gli indirizzi IP DR locali sono specificati nel file `rdqm.ini`, gli indirizzi IP DR remoti possono essere specificati sulla riga comandi:

```
crtmqm -sx -rr p -ri 192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

Se non viene specificato alcun indirizzo IP DR nel file `rdqm.ini`, è possibile specificare sia gli indirizzi IP DR remoti che quelli locali sulla riga comandi:

```
crtmqm -sx -rr p -rl 192.168.1.1,192.168.1.2,192.168.1.3 -ri
192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

L'output in risposta alla creazione di QM1 è il seguente:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'main-bob'.
Secondary queue manager created on 'main-charlie'.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qm1/qmgr/qm1' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QM1'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA secondary queue manager:
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

Copiare il comando dal messaggio per creare l'istanza secondaria DR di QM1 su dr-alice:

```
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

Il seguente messaggio viene emesso su dr-alice:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr-bob'.
Secondary queue manager created on 'dr-charlie'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
```

Verifica il DR secondario

Per verificare le funzioni di ripristino di emergenza di QM1, eseguire il seguente comando su main - alice per rendere QM1 l'istanza secondaria DR:

```
rdqmdr -m QM1 -s
Queue manager 'QM1' has been made the DR secondary on this node.
```

Eeguire il seguente comando su dr-alice per rendere QM1 l'istanza primaria DR su tale nodo:

```
rdqmdr -m QM1 -p
Queue manager 'QM1' has been made the DR primary on this node.
```

Eliminazione di un RDQM DR/HA

Per eliminare il DR/HA RDQM denominato QM1, terminare prima il gestore code su main - alice:

```
endmqm -w QM1
Replicated data queue manager disabled.
Waiting for queue manager 'QM1' to end.
IBM MQ queue manager 'QM1' ended.
```

Quindi, eseguire il seguente comando su main - alice per eliminare QM1:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'main-bob'.
Secondary queue manager deleted on 'main-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

Infine, è necessario eliminare QM1 su dr-alice:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'dr-bob'.
Secondary queue manager deleted on 'dr-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

Concetti correlati

“Funzionamento in un ambiente DR/HA” a pagina 659

Quando si opera in un ambiente DR/HA ci sono considerazioni separate per l'alta disponibilità e il ripristino di emergenza.

Attività correlate

“Creazione di RDQM DR/HA” a pagina 644

Utilizzare il comando `crtmqm` per creare un gestore code di dati replicati (RDQM) in una configurazione DR/HA.

“Eliminazione di un RDQM DR/HA” a pagina 648

Utilizzare il comando `dltmqm` per eliminare un gestore code di dati replicati DR/HA (RDQM).

CP4I MQ Adv. HA nativa

La HA nativa è una soluzione alta disponibilità disponibile sulle distribuzioni del contenitore di IBM MQ.

Una configurazione della HA nativa è composta da tre nodi (che possono essere, ad esempio, tre pod Kubernetes), ciascuno con un'istanza del gestore code. Un'istanza è il gestore code attivo, che elabora i messaggi e scrive nel relativo log. Ogni volta che il log viene scritto, il gestore code attivo invia i dati alle altre due istanze, note come 'repliche'. Ogni replica scrive nel proprio log, riconosce i dati e quindi aggiorna i propri dati della coda dal log replicato. Se il nodo su cui è in esecuzione il gestore code attivo ha esito negativo, una delle istanze di replica del gestore code assume il ruolo attivo e dispone di dati correnti con cui operare.

Per una panoramica dettagliata, vedi [HA nativa](#) nella sezione Contenitori di questa documentazione.

La seguente figura mostra una tipica distribuzione con tre istanze di un gestore code distribuite in tre contenitori.

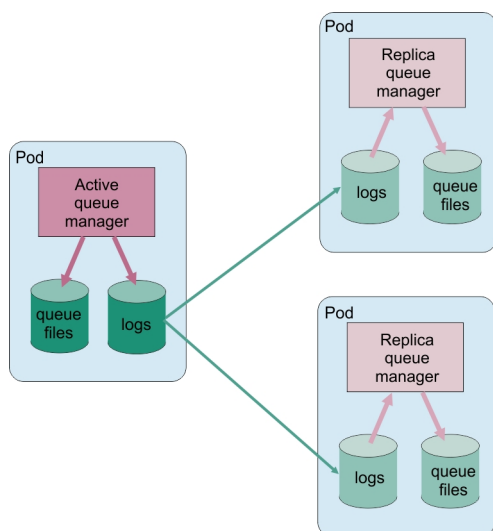


Figura 83. Esempio di configurazione della HA nativa

CP4I MQ Adv. Creazione della soluzione HA nativa

Il metodo consigliato per la creazione di una soluzione HA nativa è utilizzando IBM MQ Operator. In alternativa, puoi creare i tuoi contenitori e configurare manualmente la HA nativa.

Nota: Queste informazioni si applicano solo agli ambienti contenitore.

Per creare una soluzione Native HA utilizzando IBM MQ Operator, consultare [Native HA](#) per una panoramica e [Esempio: configurazione di un gestore code Native HA](#) per istruzioni dettagliate.

Per creare i tuoi propri contenitori e configurare manualmente la HA nativa, vedi [Creazione del gruppo della HA nativa se stai creando i tuoi propri contenitori](#).

CP4I

MQ Adv.

Chiusura gestori code della HA nativa

Per IBM MQ nei contenitori, è possibile utilizzare il comando **endmqm** per terminare un gestore code attivo o di replica che fa parte di un gruppo HA nativo.

Informazioni su questa attività

Nota: Queste informazioni si applicano solo agli ambienti contenitore.

La procedura per arrestare un gestore code che fa parte di un gruppo HA nativo dipende dal fatto che sia un'istanza di replica o attiva. Quando si termina uno dei tipi di istanza, viene eseguito un controllo per garantire che la fine dell'istanza non interrompa il quorum del gruppo HA nativo. Se il quorum viene interrotto, il comando **endmqm** ha esito negativo.

Quando si immette un comando **endmqm**, le altre istanze nel gruppo vengono avvisati che ciò si sta verificando, in modo che non riportino errori quando la connessione si interrompe.

Se un'istanza attiva perde il quorum a causa della chiusura o della disconnessione di un numero eccessivo di istanze di replica, l'istanza attiva attende per un periodo di tempo configurabile prima di terminare completamente. Ciò consente un periodo di tempo per arrestare l'elaborazione in modo normale, piuttosto che le applicazioni che hanno le connessioni interrotte. Questo valore di timeout può essere specificato dall'attributo `QuorumConnectivityTimeout` nella stanza `NativeHALocalInstance` del file `qm.ini`. Il valore predefinito è 0 secondi.

Procedura

- Per terminare l'istanza attiva di un gestore code, immettere il comando riportato di seguito sul nodo su cui è in esecuzione l'istanza attiva:

```
endmqm -s QMgrName
```

- Specificare l'opzione `-r` per aiutare le applicazioni client a riconnettersi ad un'altra istanza.
- Se questa istanza non è l'istanza attiva nel gruppo HA nativo, il comando ha esito negativo.
- Se la chiusura di questa istanza attiva causerebbe l'esito negativo del quorum del gruppo, il comando non riesce. (Se altre istanze terminano o diventano non disponibili contemporaneamente all'esecuzione di questo comando, il controllo del quorum potrebbe non rilevarlo, il gruppo HA nativo termina e può essere riavviato solo quando è disponibile un numero sufficiente di istanze).

Quando il gestore code attivo termina, una delle istanze di replica assume il ruolo attivo. Non è possibile specificare quale replica assume il controllo, ciò è determinato dalla negoziazione all'interno del gruppo e dipende da quale ha i log di transazione più aggiornati.

- Per terminare un'istanza di replica di un gestore code, immettere il seguente comando:

```
endmqm -x QMgrName
```

- Se questa è l'istanza attiva, il comando ha esito negativo.
- Se la chiusura di questa istanza di replica causa l'esito negativo del quorum del gruppo, il comando ha esito negativo. (Se altre istanze terminano o diventano non disponibili contemporaneamente all'esecuzione di questo comando, il controllo del quorum potrebbe non rilevarlo, il gruppo HA nativo termina e può essere riavviato solo quando è disponibile un numero sufficiente di istanze).

Nota: È anche possibile utilizzare gli switch `-c`, `-i`, `-p` o `-w` con il comando **endmqm** sulle istanze della HA nativa, indipendentemente dal ruolo in cui si trovano. L'istanza del gestore code viene terminata, ignorando l'effetto che ha sul quorum del gruppo. Tuttavia, le informazioni sono ancora condivise con

le altre istanze del gruppo. È possibile utilizzare questi switch insieme a -s per l'istanza attiva. Non è possibile utilizzare questi switch insieme allo switch -x per istanze di replica.

Riferimenti correlati

[endmqm \(fine gestore code\)](#)

Registrazione: verifica che i messaggi non vengano persi

IBM MQ registra tutte le modifiche significative ai dati permanenti controllati dal gestore code in un log di ripristino.

Ciò include la creazione e l'eliminazione di oggetti, aggiornamenti di messaggi persistenti, stati delle transazioni, modifiche agli attributi degli oggetti e attività del canale. Il file di log contiene le informazioni necessarie per ripristinare tutti gli aggiornamenti alle code di messaggi mediante:

- Conservazione dei record delle modifiche del gestore code
- Conservazione dei record degli aggiornamenti della coda per l'utilizzo da parte del processo di riavvio
- Abilitazione al ripristino dei dati dopo un errore hardware o software

Tuttavia, IBM MQ si basa anche sul sistema disco che ospita i file, inclusi i file di log. Se il sistema disco è esso stesso inaffidabile, le informazioni, incluse le informazioni di log, possono ancora essere perse.



Avvertenza: Non è possibile spostare i log di ripristino su un sistema operativo differente.

Aspetto dei log

I log sono costituiti da file primari e secondari e un file di controllo. Si definisce il numero e la dimensione dei file di log e la posizione in cui vengono memorizzati nel file system.

Un log IBM MQ è composto da due componenti:

1. Uno o più file di dati di log.
2. Un file di controllo log

Un file di dati di log è noto anche come estensione di log.

Esistono diverse estensioni di log che contengono i dati registrati. È possibile definirne il numero e la dimensione (come spiegato in “Stanza `LogDefaults` del file `mqs.ini`” a [pagina 102](#)) oppure utilizzare il valore predefinito di sistema di tre estensioni primarie e due secondarie.

Ciascuna delle tre estensioni primaria e secondaria assume il valore predefinito di 16 MB.

Quando si crea un gestore code, il numero di estensioni log preassegnate è il numero di estensioni log *primarie* assegnate. Se non si specifica un numero, viene utilizzato il valore predefinito.

IBM MQ utilizza due tipi di registrazione:

- Circolare
- Lineare

Il numero di estensioni di log utilizzate con la registrazione lineare può essere molto elevato, a seconda della frequenza della registrazione dell'immagine multimediale.

Per ulteriori informazioni, consultare “[Tipi di registrazione](#)” a [pagina 665](#).



Nei sistemi IBM MQ for AIX or Linux , se non è stato modificato il percorso di log, le estensioni di log vengono create nella directory:

```
/var/mqm/log/QMgrName
```


Windows In IBM MQ for Windows, se il percorso di log non è stato modificato, le estensioni di log vengono create nella directory:

```
C:\ProgramData\IBM\MQ\Log\QMGrName
```

IBM MQ inizia con queste estensioni di log primarie, ma se lo spazio di log primario non è sufficiente, alloca le estensioni di log *secondarie*. Lo fa dinamicamente e li rimuove quando la richiesta di spazio di log si riduce. Per impostazione predefinita, è possibile assegnare fino a due estensioni di log secondarie. È possibile modificare questa allocazione predefinita, come descritto in [“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms”](#) a pagina 93.

Le estensioni log sono precedute dalla lettera S o dalla lettera R. Le estensioni attive, inattive e superflue hanno come prefisso S, mentre le estensioni di riutilizzo hanno come prefisso R.

Quando si esegue il backup o il ripristino del gestore code, eseguire il backup e il ripristino di tutte le estensioni attive, inattive e superflue, insieme al file di controllo log.

Nota: Non è necessario eseguire il backup e ripristinare le estensioni di riutilizzo.

Il file di controllo log

Il file di controllo log contiene le informazioni necessarie per descrivere lo stato delle estensioni di log, ad esempio dimensione e ubicazione, e il nome della successiva estensione disponibile.

Importante: Il file di controllo log è solo per uso interno del gestore code.

Il gestore code conserva i dati di controllo associati con lo stato del log di ripristino nel file di controllo log e non è necessario modificare il contenuto del file di controllo log.

Il file di controllo log si trova nel percorso di log ed è denominato `amqh1ctl.lfh`. Quando si esegue il backup o il ripristino del gestore code, assicurarsi che venga eseguito il backup e il ripristino del file di controllo log, insieme alle estensioni di log.

Tipi di registrazione

In IBM MQ sono disponibili due modalità di gestione dei record delle attività del gestore code: registrazione circolare e registrazione lineare. Un terzo tipo di registrazione, replicato, viene utilizzato solo dalle configurazioni della HA nativa.

registrazione circolare

Utilizzare la registrazione circolare se si desidera solo riavviare il ripristino, utilizzando il log per eseguire il rollback delle transazioni che erano in corso quando il sistema è stato arrestato.

La registrazione circolare mantiene tutti i dati di riavvio in un anello di file di log. La registrazione completa il primo file dell'anello e quindi passa al successivo e così via, fino a quando tutti i file sono completi. Successivamente, torna al primo file dell'anello e ricomincia. Questo processo va avanti finché il prodotto è in uso e comporta il vantaggio che non si rimane mai senza file di log durante l'esecuzione.

IBM MQ conserva le voci di log richieste per riavviare il gestore code senza perdita di dati fino a quando non sono più necessarie per garantire il recupero dei dati del gestore code. Il meccanismo per rilasciare i file di log per il riutilizzo è descritto in [“Utilizzo del punto di controllo per garantire il ripristino completo”](#) a pagina 667.

registrazione lineare

Utilizzare la registrazione lineare se si desidera riavviare il ripristino e il ripristino del supporto (ricreando i dati persi o danneggiati riproducendo il contenuto del log). La registrazione lineare conserva i dati di log in una sequenza continua di file di log.

I file di log possono essere:

- Riutilizzati, ma solo quando non sono più necessari per il ripristino del riavvio o del supporto.

- Archiviato manualmente per analisi e storage a lungo termine.

La frequenza delle immagini dei supporti determina quando i file di log lineari possono essere riutilizzati ed è un fattore importante nella quantità di spazio su disco che deve essere disponibile per i file di log lineari.

È possibile configurare il gestore code per eseguire automaticamente le immagini dei supporti periodici, in base all'ora o all'utilizzo del log, oppure è possibile pianificare manualmente le immagini dei supporti.

L'amministratore decide la politica da implementare e le implicazioni sull'utilizzo dello spazio su disco. I file di log necessari per il ripristino del riavvio devono essere sempre disponibili, mentre i file di log necessari solo per il ripristino del supporto possono essere archiviati in una memoria a più lungo termine, ad esempio nastro.

Se l'amministratore abilita la gestione automatica dei log e le immagini dei supporti automatici, la registrazione lineare si comporta in modo simile a un log circolare molto grande, ma con la ridondanza migliorata contro i malfunzionamenti dei supporti abilitata dal ripristino dei supporti.

È possibile modificare un tipo di log esistente per un gestore code, da lineare a circolare o da circolare a lineare utilizzando il comando `migmqlog`.

Registrazione replicata

CP4I

Utilizzare la registrazione replicata per configurare una configurazione della HA nativa. Quando si crea un gruppo HA nativo, vengono creati tre gestori code su nodi differenti. Si specifica un tipo di registrazione replicata insieme a un nome istanza univoco per ciascuno dei gestori code. La configurazione della HA nativa fornisce una soluzione di alta disponibilità avendo un'istanza attiva che replica i dati di log su due istanze di replica. Se l'istanza attiva ha esito negativo, una delle istanze di replica assume il ruolo attivo. La replica del log garantisce che i dati, se presenti, vengano persi. Per ulteriori dettagli, vedere [“HA nativa” a pagina 662](#). Un log replicato è equivalente a un log lineare con la gestione log automatica e le immagini dei supporti automatici abilitate.

Estensioni log lineari non attive

Multi

Se si utilizza la gestione log automatica, inclusa l'archiviazione, il programma di registrazione tiene traccia delle estensioni log lineari non attive.



Attenzione: Se si sta utilizzando la gestione automatica dei log, senza archiviazione, l'utilizzo di un gestore code di backup non è supportato per questo processo.

ALW

Quando un'estensione di log non è più richiesta per il recupero e, se necessario, viene archiviata, il programma di registrazione eliminerà l'estensione di log o la riutilizzerà.

Un'estensione di log riutilizzata viene ridenominata per essere la successiva nella sequenza di log. Viene scritto periodicamente il messaggio AMQ7490, che indica quante estensioni sono state create, eliminate o riutilizzate.

Il programma di registrazione sceglie il numero di estensioni da mantenere pronte per il riutilizzo e quando eliminare tali estensioni.

Log attivo

Esistono diversi file che si dice siano *attivi* sia nella registrazione lineare che nella registrazione circolare. Il log attivo è la quantità massima di spazio di log, indipendentemente dal fatto che si stia utilizzando la registrazione circolare o lineare, a cui potrebbe fare riferimento il ripristino di riavvio.

Il numero di file di log attivi è generalmente inferiore al numero di file di log primari come definito nei file di configurazione. (Consultare [“Calcolo della dimensione del log” a pagina 671](#) per informazioni sulla definizione del numero.)

Si noti che lo spazio di log attivo non include lo spazio richiesto per il ripristino del supporto e che il numero di file di log utilizzati con la registrazione lineare può essere molto grande, a seconda del flusso di messaggi e della frequenza delle immagini del supporto.

Log inattivo

Quando un file di log non è più necessario per riavviare il ripristino, diventa *inattivo*. I file di log che non sono richiesti per riavviare il ripristino o il ripristino del supporto, possono essere considerati come file di log superflui.

Quando si utilizza la gestione log automatica, il gestore code controlla l'elaborazione di questi file di log superflui. Se è stata selezionata la gestione manuale del log, diventa responsabilità dell'amministratore gestire (ad esempio, eliminare e archiviare) i file di log superflui se non sono più di interesse per l'operazione.

Fare riferimento a [“Gestione dei log” a pagina 677](#) per ulteriori informazioni sulla disposizione dei file di log.

File di log secondari

Sebbene i file di log secondari siano definiti per la registrazione lineare, non vengono utilizzati nelle normali operazioni. Se si verifica una situazione quando, probabilmente a causa di transazioni di lunga durata, non è possibile liberare un file dal pool attivo perché potrebbe essere ancora richiesto per un riavvio, i file secondari vengono formattati e aggiunti al pool di file di log attivo.

Se il numero di file secondari disponibili viene utilizzato, le richieste per la maggior parte delle ulteriori operazioni che richiedono l'attività di log verranno rifiutate con un codice di ritorno MQRC_RESOURCE_PROBLEM restituito all'applicazione e qualsiasi transazione di lunga durata verrà considerata per il rollback asincrono.



Attenzione: Tutti i tipi di registrazione possono far fronte a una perdita di alimentazione imprevista, supponendo che non vi sia alcun malfunzionamento hardware.

Utilizzo del punto di controllo per garantire il ripristino completo

I gestori code di registrazione circolare e di registrazione lineare supportano il ripristino del riavvio. Indipendentemente da quanto bruscamente l'istanza precedente del gestore code viene terminata (ad esempio, un'interruzione dell'alimentazione) al riavvio, il gestore code ripristina il suo stato persistente allo stato transazionale corretto al momento della chiusura.

Il ripristino del riavvio dipende dal mantenimento dell'integrità del disco. Allo stesso modo, il sistema operativo deve garantire l'integrità del disco indipendentemente da quanto bruscamente potrebbe verificarsi la chiusura di un sistema operativo.

Nell'evento altamente insolito in cui l'integrità del disco non viene mantenuta, la registrazione lineare (e il ripristino dei supporti) fornisce ulteriori opzioni di ridondanza e recuperabilità. Con una tecnologia sempre più comune, come RAID, è sempre più raro soffrire di problemi di integrità del disco e molte aziende configurano la registrazione circolare e utilizzano solo il ripristino di riavvio.

IBM MQ è progettato come un classico gestore risorse Write Ahead Logging. Gli aggiornamenti persistenti alle code messaggi si verificano in due fasi:

1. I record di log che rappresentano l'aggiornamento vengono scritti in maniera affidabile nel log di ripristino
2. Il file o i buffer della coda vengono aggiornati in un modo che è il più efficiente per il sistema, ma non necessariamente coerente.

I file di log possono quindi diventare più aggiornati del buffer della coda sottostante e dello stato del file.

Se a questa situazione è stato consentito di continuare senza interruzione, è necessario un volume molto elevato di ripetizione del log per rendere congruente lo stato della coda in seguito a un ripristino da un arresto anomalo.

IBM MQ utilizza checkpoints per limitare il volume di ripetizione del log richiesto in seguito a un ripristino da un arresto anomalo. L'evento chiave che controlla se un file di log è definito attivo o meno è un checkpoint.

Un punto di controllo IBM MQ è un punto:

- Di congruenza tra il log di recupero e i file oggetto.
- Ciò identifica una posizione nel log, da cui è garantita la ripetizione di inoltro dei record di log successivi per ripristinare la coda allo stato logico corretto nel momento in cui il gestore code potrebbe essere terminato.

Durante un punto di controllo, IBM MQ elimina i vecchi aggiornamenti ai file delle code, come richiesto, per limitare il volume dei record di log che devono essere riprodotti per riportare le code ad uno stato congruente dopo un recupero da un arresto anomalo.

Il punto di controllo completo più recente contrassegna un punto nel log da cui deve essere eseguita la ripetizione durante il recupero da caduta. La frequenza del checkpoint è quindi un compromesso tra il sovraccarico dei checkpoint di registrazione e il miglioramento del tempo di recupero potenziale implicito da tali checkpoint.


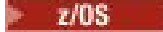
Il programma di registrazione pianifica i punti di controllo più frequentemente (in modo che il successivo venga pianificato prima del completamento del precedente) poiché il programma di registrazione sta tentando di mantenere il log attivo nelle estensioni del log primario. Se ciò non è possibile, viene registrato un errore [AMQ7466](#).

La posizione nel log dell'inizio del punto di controllo completo più recente è uno dei fattori chiave per determinare se un file di log è attivo o inattivo. L'altro fattore chiave è la posizione nel log del primo record di log relativo al primo aggiornamento persistente effettuato da una transazione attiva corrente.

Se un nuovo punto di controllo viene registrato nel secondo file di log o in quello successivo e nessuna transazione corrente fa riferimento a un record di log nel primo file di log, il primo file di log diventa inattivo. Nel caso di registrazione circolare, il primo file di log è ora pronto per essere riutilizzato. In caso di registrazione lineare, il primo file di log sarà in genere ancora richiesto per il ripristino del supporto.


Se si configura la registrazione circolare o la gestione log automatica, il gestore code gestirà i file di log inattivi. Se si configura la registrazione lineare con la gestione manuale dei log, questa diventa un'attività amministrativa per gestire i file inattivi in base ai requisiti dell'operazione.

IBM MQ genera automaticamente i punti di controllo. Essi vengono presi nei seguenti orari:

- Quando il gestore code viene avviato
- Alla chiusura
- Quando lo spazio di registrazione è insufficiente
-  **Multi** Dopo che sono state registrate 50.000 operazioni dal momento in cui è stato eseguito il punto di controllo precedente
-  **z/OS** Dopo che *number_of_operations* è stato registrato da quando è stato eseguito il punto di controllo precedente, dove *number_of_operations* è il numero di operazioni impostato nella proprietà **LOGLOAD**.

Quando IBM MQ viene riavviato, trova l'ultimo record di checkpoint nel log. Queste informazioni vengono conservate nel file del punto di controllo aggiornato alla fine di ogni punto di controllo. Tutte le operazioni che hanno avuto luogo dopo il checkpoint vengono rieseguite in avanti. Questa è nota come fase di ripetizione.

La fase di ripetizione riporta le code allo stato logico in cui si trovavano prima dell'errore o della chiusura del sistema. Durante la fase di ripetizione, viene creato un elenco delle transazioni che erano in corso quando si è verificato l'errore di sistema o la chiusura.

 **Multi** I messaggi [AMQ7229](#) e [AMQ7230](#) vengono emessi per indicare l'avanzamento della fase di ripetizione.

Per sapere quali operazioni eseguire il backout o il commit, IBM MQ accede a ciascun record di log attivo associato a una transazione incompleta. Questa è nota come fase di recupero.

Multi Vengono emessi i messaggi [AMQ7231](#), [AMQ7232](#) e [AMQ7234](#) per indicare l'avanzamento della fase di ripristino.

Una volta eseguito l'accesso a tutti i record di log necessari durante la fase di recupero, ciascuna transazione attiva viene a sua volta risolta e ciascuna operazione associata alla transazione verrà sottoposta a backout o a commit. Questa è nota come la fase di risoluzione.

Multi Viene emesso un messaggio [AMQ7233](#) per indicare l'avanzamento della fase di risoluzione.

z/OS Su z/OS, il riavvio dell'elaborazione è composto da varie fasi.

1. L'intervallo di registrazione per il recupero viene stabilito, in base al recupero del supporto richiesto per le serie di pagine e al record di registrazione meno recente richiesto per il ripristino delle unità di lavoro e per l'ottenimento dei blocchi per le unità di lavoro in dubbio.
2. Una volta determinato l'intervallo di log, la lettura del log in avanti viene eseguita per portare le serie di pagine allo stato più recente e anche per bloccare i messaggi relativi alle unità di lavoro in dubbio o in corso.
3. Quando la lettura del log di inoltro è stata completata, i log vengono letti all'indietro per eseguire il backout di tutte le unità di lavoro che erano in fase di elaborazione o in fase di backout al momento dell'errore.

z/OS Un esempio dei messaggi che potrebbero essere visualizzati:

```
CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS - 806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range 815
from RBA=000000001E45FF7AD to RBA=000000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE - 816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range 817
from RBA=000000001E48C1882 to RBA=000000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE - 818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED
```

Nota: Se c'è una grande quantità di log da leggere, i messaggi CSQR031I (recupero inoltro) e CSQR033I (recupero all'indietro) vengono emessi periodicamente per mostrare l'avanzamento.

In [Figura 84 a pagina 670](#), tutti i record precedenti all'ultimo checkpoint, Checkpoint 2, non sono più necessari per IBM MQ. Le code possono essere recuperate dalle informazioni del punto di controllo e da eventuali voci di log successive. Per la registrazione circolare, è possibile riutilizzare tutti i file liberati prima del punto di controllo. Per un log lineare, non è più necessario accedere ai file di log liberati per le normali operazioni e diventare inattivi. Nell'esempio, il puntatore di testa della coda viene spostato in modo da puntare all'ultimo punto di controllo, Checkpoint 2, che diventa la nuova testa della coda, Head 2. Il file di log 1 può ora essere riutilizzato.

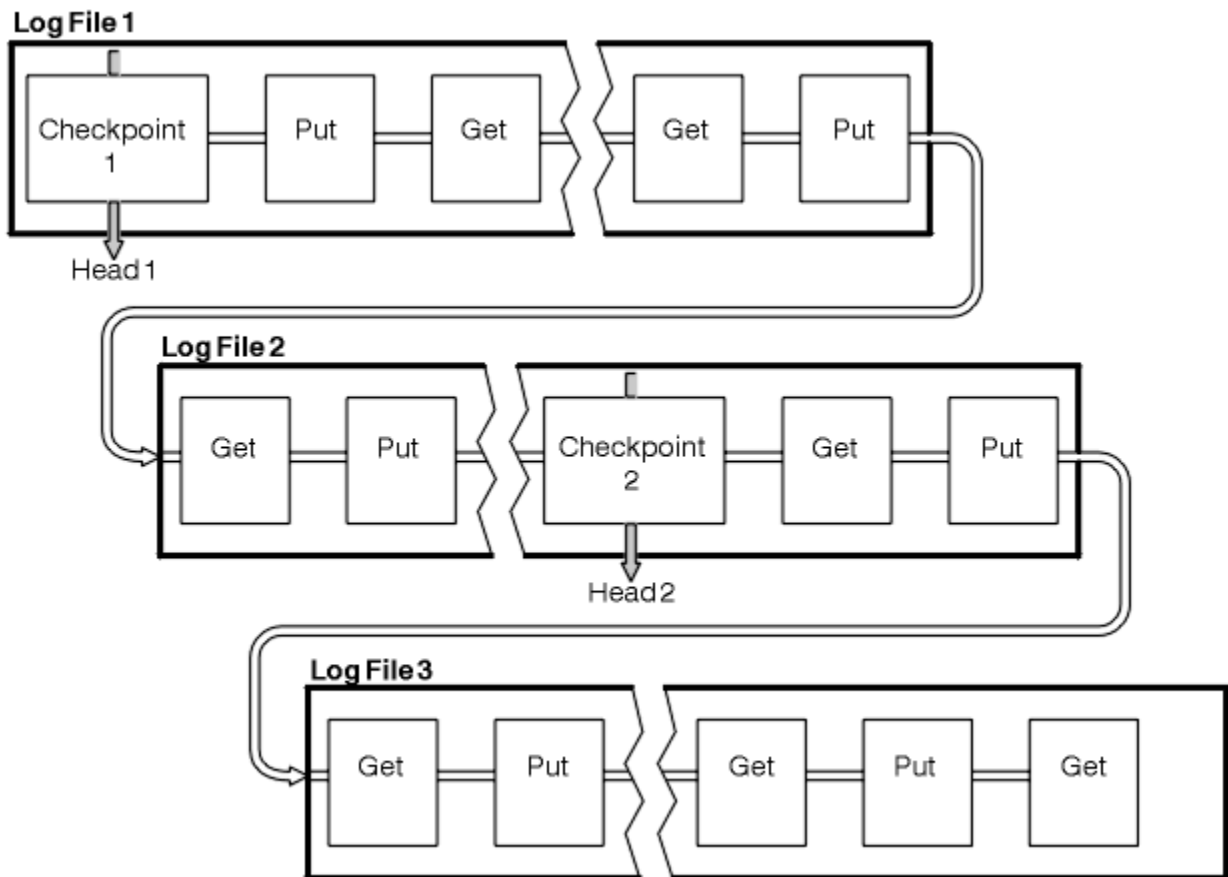


Figura 84. Checkpoint

Checkpoint con transazioni di lunga durata

Come una transazione di lunga durata influisce sul riutilizzo dei file di log.

Figura 85 a pagina 671 mostra in che modo una transazione di lunga durata influisce sul riutilizzo dei file di log. Nell'esempio, una transazione di lunga durata ha creato una voce nel log, mostrata come LR 1, dopo il primo checkpoint visualizzato. La transazione non viene completata (al punto LR 2) fino a dopo il terzo punto di controllo. Tutte le informazioni di log da LR 1 in poi vengono conservate per consentire il ripristino di tale transazione, se necessario, fino al completamento.

Una volta completata la transazione di lunga durata, a LR 2, la parte principale del log si sposta logicamente al Checkpoint 3, l'ultimo checkpoint registrato. I file che contengono i record di log prima del punto di controllo 3, intestazione 2, non sono più necessari. Se si utilizza la registrazione circolare, lo spazio può essere riutilizzato.

Se i file di log primari sono completamente pieni prima del completamento della transazione di lunga durata, i file di log secondari potrebbero essere utilizzati per evitare che i log si riempiano.

Le attività che sono interamente sotto il controllo del gestore code, ad esempio il punto di controllo, sono pianificate per provare a mantenere l'attività nel log primario.

Tuttavia, quando è richiesto lo spazio di log secondario per supportare il comportamento al di fuori del controllo del gestore code (ad esempio, la durata di una delle transazioni), il gestore code tenta di utilizzare qualsiasi spazio di log secondario definito, per consentire il completamento di tale attività.

Se tale attività non viene completata entro il periodo di tempo in cui l'80% dello spazio di log totale è in uso, il gestore code avvia l'azione per recuperare spazio di log, indipendentemente dal fatto che ciò abbia un impatto sull'applicazione.

Quando l'intestazione del log viene spostata e si utilizza la registrazione circolare, i file di log primari potrebbero diventare idonei per il riutilizzo e il programma di registrazione, dopo aver riempito il file corrente, riutilizza il primo file primario disponibile. Se si utilizza la registrazione lineare, l'intestazione

del log viene ancora spostata verso il basso nel pool attivo e il primo file diventa inattivo. Un nuovo file primario viene formattato e aggiunto alla fine del lotto in modo da essere pronto per future attività di registrazione.

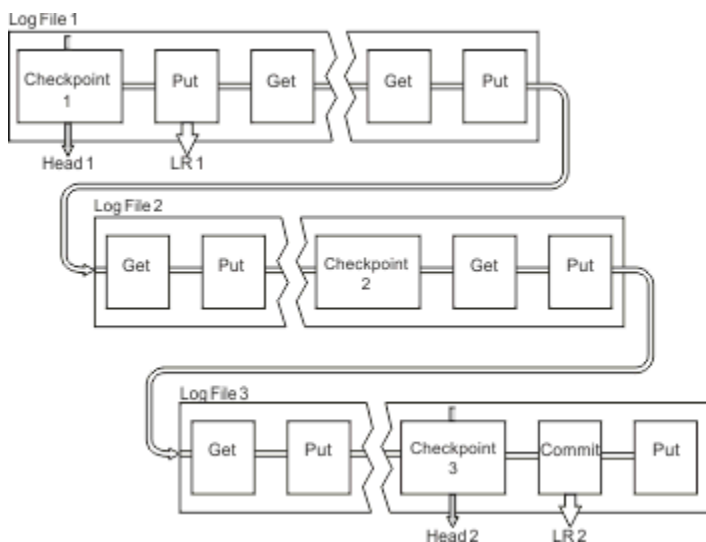


Figura 85. Checkpoint con una transazione di lunga durata

Calcolo della dimensione del log

Stima della dimensione del log necessaria per un gestore code.

Dopo aver deciso se il gestore code utilizza la registrazione circolare o lineare, è necessario stimare la dimensione del Log attivo richiesto dal gestore code. La dimensione del log attivo è determinata dai seguenti parametri di configurazione del log:

LogFilePages

La dimensione di ciascun file di log primario e secondario in unità di pagine 4K

LogPrimaryFiles

Il numero di file di log primari preassegnati

LogSecondaryFiles

Il numero di file di log secondari che possono essere creati per l'utilizzo quando i file di log primari stanno diventando pieni

Note:

1. È possibile modificare il numero di file di log primari e secondari ogni volta che il gestore code viene avviato, anche se non è possibile notare immediatamente l'effetto della modifica apportata ai log secondari.
2. Non è possibile modificare la dimensione del file di log; è necessario determinarlo **prima** di creare il gestore code.
3. Il numero di file di log primari e la dimensione del log determinano la quantità di spazio di log preassegnato quando viene creato il gestore code.
4. Il numero totale di file di log primari e secondari non può superare 511 su sistemi AIX and Linux o 255 su Windows, che in presenza di transazioni di lunga durata limita la quantità massima di spazio di log disponibile per il gestore code per il ripristino del riavvio. La quantità di spazio di log di cui il gestore code potrebbe aver bisogno per il ripristino del supporto non condivide questo limite.
5. Quando viene utilizzata la registrazione *circolare*, il gestore code riutilizza lo spazio di log primario e secondario. Il gestore code assegnerà, fino a un limite, un file di log secondario quando un file di log diventa pieno e il successivo file di log primario nella sequenza non è disponibile.

Consultare [“Quanto deve essere grande il mio log attivo?”](#) a pagina 672 per informazioni sul numero di log da assegnare. Le estensioni di log principali vengono utilizzate in sequenza e tale sequenza non cambia.

Ad esempio, se hai tre log principali 0, 1 e 2, l'ordine di utilizzo è 0,1,2 seguito da 1,2,0, 2,0,1, torna a 0,1,2 e così via. Tutti i log secondari assegnati vengono intervallati come richiesto.

6. I file di log primari vengono resi disponibili per il riutilizzo durante un checkpoint. Il gestore code prende in considerazione sia lo spazio di log primario che quello secondario prima di prendere in considerazione un punto di controllo perché la quantità di spazio di log è insufficiente.

Il gestore code tenta di pianificare i checkpoint in un modo che mantiene l'utilizzo del log all'interno delle estensioni primarie.

Per ulteriori informazioni, consultare [“Stanza LogDefaults del file mq5.ini” a pagina 102](#).

Quanto deve essere grande il mio log attivo?

Stima della dimensione del log attivo necessaria per un gestore code.

La dimensione del log attivo è limitata da:

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

Il log dovrebbe essere abbastanza grande da gestire la transazione in esecuzione da più tempo quando il gestore code scrive la quantità massima di dati al secondo sul disco.

Se la transazione in esecuzione da più tempo per N secondi e la quantità massima di dati al secondo scritti sul disco dal gestore code è di B byte al secondo nel log, il log deve essere almeno:

```
logsize >= 2 * (N+1) * B
```

È probabile che il gestore code stia scrivendo la quantità massima di dati al secondo sul disco quando si sta eseguendo il carico di lavoro di picco o quando si stanno registrando le immagini dei supporti.

Se una transazione viene eseguita per un periodo di tempo tale che l'estensione del log contenente il primo record di log non è contenuta nel log attivo, il gestore code esegue il rollback delle transazioni attive una alla volta, a partire dalla transazione con il record di log più vecchio.

Il gestore code deve rendere inattive le vecchie estensioni di log prima che venga utilizzato il numero massimo di file primari e secondari e il gestore code deve assegnare un'altra estensione di log.

Decidere per quanto tempo si desidera eseguire la transazione con esecuzione più lunga, prima che al gestore code sia consentito eseguire il rollback. La transazione con esecuzione più lunga potrebbe essere in attesa di un traffico di rete lento o, nel caso di una transazione mal progettata, in attesa di input dell'utente.

È possibile analizzare il tempo di esecuzione della transazione più lungo, immettendo il seguente comando **runmqsc** :

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

L'immissione del comando `dspmqtin -a` mostra tutti i comandi XA e non XA in tutti gli stati.

L'immissione di questo comando elenca la data e l'ora in cui è stato scritto il primo record di log per tutte le transazioni correnti.



Attenzione: Ai fini del calcolo della dimensione del log, è l'ora in cui è stato scritto il primo record di log che conta, non l'ora in cui è stata avviata l'applicazione o la transazione. Arrotondare la lunghezza della transazione con esecuzione più lunga al secondo più vicino. Ciò è dovuto alle ottimizzazioni nel gestore code.

Il primo record di log può essere scritto molto tempo dopo l'avvio dell'applicazione, se l'applicazione inizia, ad esempio, con l'emissione di una chiamata MQGET che attende per un periodo di tempo prima di ricevere effettivamente un messaggio.

Esaminando l'output massimo di data e ora osservato dal

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```


il comando immesso in origine, dalla data e ora correnti, è possibile stimare per quanto tempo viene eseguita la transazione in esecuzione più a lungo.

Assicurarsi di eseguire questo comando **runmqsc** ripetutamente mentre le transazioni con esecuzione più lunga sono in esecuzione nel carico di lavoro di picco in modo da non sottovalutare la lunghezza della transazione con esecuzione più lunga.

In IBM MQ 8.0 utilizzare gli strumenti del sistema operativo, ad esempio **iostat** su piattaforme UNIX .

È possibile rilevare i byte al secondo che il gestore code sta scrivendo nel log immettendo il seguente comando:

```
amqsrua -m qmgr -c DISK -t Log
```

I byte logici scritti, mostrano i byte al secondo scritti dal gestore code nel log. Ad esempio:

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

In questo esempio, i byte logici al secondo scritti nel log sono 12760669/sec o circa 12 MiB al secondo.

Utilizzo di

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

ha mostrato che la transazione con esecuzione più lunga era:

```
CONN(57E14F6820700069)
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r) UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```

Poiché la data e l'ora correnti erano 2016-09-20 16.44.19, questa transazione era in esecuzione da 5 secondi. Tuttavia, è necessario tollerare le transazioni in esecuzione per 10 secondi prima che il gestore code esegua il rollback. Quindi la dimensione del log dovrebbe essere:

```
2 * (10 + 1) * 12 = 264 MiB
```

.

Il numero di file di log deve essere in grado di contenere la dimensione di log massima prevista (calcolata nel testo precedente). Questo sarà:

Numero minimo di file di log = (dimensione log richiesta) / (**LogFilePages** * dimensione pagina file di log (4096))

Utilizzando il valore predefinito **LogFilePages**, che è 4096, e la stima della dimensione del log di 264MiB, calcolata nel testo precedente, il numero minimo di file di log dovrebbe essere:

```
264MiB / (4096 x 4096) = 16.5
```

cioè 17 file di log.

Se si dimensiona il log in modo che il carico di lavoro previsto venga eseguito all'interno dei file primari:

- I file secondari forniscono una certa contingenza nel caso in cui sia necessario ulteriore spazio di log.
- La registrazione circolare utilizza sempre i file primari preassegnati, che è marginalmente più veloce dell'allocazione e della deallocazione dei file secondari.

- Il gestore code utilizza solo lo spazio rimanente nei file primari per calcolare quando eseguire il successivo checkpoint.

Pertanto, nell'esempio precedente, impostare i seguenti valori in modo che il workload venga eseguito all'interno dei file di log primari:

- **LogFilePages** = 4096
- **LogPrimaryFiles** = 17
- **LogSecondaryFiles** = 5

Tieni presente quanto segue:

- In questo esempio, 5 secondario è più del 20% dello spazio di log attivo.

Il programma di registrazione tenta di mantenere il carico di lavoro nei soli file primari. Di conseguenza, il programma di registrazione pianifica i punti di controllo quando una frazione dei soli file primari è piena.

Avere i file secondari è una contingenza, nel caso in cui ci siano transazioni di esecuzione inaspettatamente lunga.

Tenere presente che il gestore code esegue un'azione per ridurre l'utilizzo dello spazio di log quando più dell' 80% dello spazio di log totale è in uso.

- Eseguire lo stesso calcolo se si utilizza la registrazione lineare o circolare.

Non fa alcuna differenza se si sta calcolando la dimensione di un log attivo lineare o circolare, poiché il concetto di log attivo significa lo stesso sia nella registrazione lineare che nella registrazione circolare.

- Le estensioni di log necessarie solo per il ripristino del supporto non si trovano all'interno del log attivo e pertanto non vengono conteggiate nel numero dei file primari e secondari.
- Il campo *LOGUTIL* di *DISPLAY QMSTATUS LOG* è disponibile per aiutarti a calcolare, approssimativamente, la dimensione del log attivo richiesto.

Questo campo è progettato per consentire all'utente di effettuare una stima ragionevole della dimensione del log richiesta senza eseguire costantemente il campionamento per determinare la durata delle transazioni più lunghe in esecuzione o la velocità di trasmissione massima del gestore code.

Quanto dovrei rendere grandi le mie pagine LogFile?

Generalmente, rendere le pagine LogFilesufficientemente grandi in modo da poter aumentare facilmente la dimensione del log attivo senza raggiungere il numero massimo di file primari. Alcuni file di log di grandi dimensioni sono preferibili a molti file di log di piccole dimensioni poiché alcuni file di log di grandi dimensioni consentono una maggiore flessibilità per aumentare la dimensione del log, se necessario.

Per la registrazione lineare, file di log molto grandi potrebbero rendere la variabile delle prestazioni. Con file di log molto grandi c'è un passo più grande per creare e formattare un nuovo file di log, o per archivarne uno vecchio. Questo è più di un problema con la gestione dei log di archivio e manuale perché con la gestione dei log automatica i nuovi file di log vengono creati raramente.

Cosa succede se il mio log è troppo piccolo?

Punti da considerare quando si stima la dimensione minima del log.

Se si rende il log troppo piccolo:

- Le transazioni di lunga durata verranno ripristinate.
- Il punto di controllo successivo deve essere avviato prima della fine del punto di controllo precedente.

Importante: Non importa quanto sia imprecisa la stima della dimensione del log, l'integrità dei dati viene mantenuta.

Consultare [“Utilizzo del punto di controllo per garantire il ripristino completo”](#) a pagina 667 per una spiegazione dei punti di controllo. Se la quantità di spazio di log rimasta nelle estensioni di log attive sta diventando breve, il gestore code pianifica i punti di controllo più frequentemente.

Un punto di controllo richiede una certa quantità di tempo; non è istantaneo. Più dati devono essere registrati nel punto di controllo, più tempo impiega il punto di controllo. Se il log è un piccolo punto di controllo può sovrapporsi, il che significa che il punto di controllo successivo è richiesto prima che il punto di controllo precedente sia terminato. Se ciò si verifica, vengono scritti dei messaggi di errore.

Se viene eseguito il backout delle transazioni di lunga durata o i punti di controllo si sovrappongono, il gestore code continua l'elaborazione del carico di lavoro. Le transazioni di breve durata continuano ad essere eseguite normalmente.

Tuttavia, il gestore code non è in esecuzione in modo ottimale e le prestazioni potrebbero essere ridotte. È necessario riavviare il gestore code con spazio di log sufficiente.

Cosa succede se il mio log è troppo grande?

Punti da considerare quando si stima la dimensione massima del log.

Se il log è troppo grande:

- È possibile aumentare il tempo impiegato per un riavvio di emergenza, anche se ciò è improbabile.
- Si sta utilizzando spazio su disco non necessario.
- Le transazioni a esecuzione molto lunga sono tollerate.

Importante: Non importa quanto sia imprecisa la stima della dimensione del log, l'integrità dei dati viene mantenuta.

Per facilitare la stima della dimensione massima del log, è possibile utilizzare le statistiche di utilizzo del log. Per ulteriori informazioni, consultare [“Decidere come impostare IMGLOGLN e IMGINTVL”](#) a pagina 681 e [ALTER QMGR](#).

Consultare [“Utilizzo del punto di controllo per garantire il ripristino completo”](#) a pagina 667 per una descrizione del modo in cui il gestore code legge il log al riavvio. Il gestore code riproduce il log dall'ultimo punto di controllo, quindi risolve tutte le transazioni che erano attive al termine del gestore code.

Per risolvere una transazione, il gestore code legge nuovamente tutti i record di log associati a tale transazione. Questi record di log potrebbero essere precedenti all'ultimo punto di controllo.

Assegnando al gestore code un log molto grande, si fornisce al gestore code l'autorizzazione a leggere ogni record di log al riavvio, anche se di solito il gestore code non deve eseguire questa operazione. Potenzialmente, nell'improbabile eventualità che ciò accada, tale processo potrebbe richiedere molto tempo.

Se il checkpoint è stato inaspettatamente arrestato prima della chiusura del gestore code, ciò aumenta notevolmente il tempo di riavvio per un gestore code con un log di grandi dimensioni. La limitazione della dimensione del log limita il tempo di riavvio di emergenza.

Per evitare questi problemi è necessario assicurarsi che:

- Il tuo carico di lavoro può adattarsi comodamente in un log che non è eccessivamente grande.
- Si evitano transazioni di lunga durata.

Quanto deve essere grande il file system del log?

Stima della dimensione del file system di log necessaria per un gestore code.

È importante che il file system del log sia sufficientemente grande, in modo che il gestore code disponga di spazio sufficiente per scrivere il log. Se il gestore code riempie completamente il filesystem di log, scriverà FFDC, eseguirà il rollback delle transazioni e potrebbe terminare improvvisamente il gestore code.

La quantità di spazio su disco riservata per il log deve essere almeno pari a quella del log attivo. Esattamente quanto più grande dipende da:

- La scelta del tipo di log (lineare o circolare)
- La dimensione del log attivo (file primari, file secondari, pagine del file di log)
- La scelta della gestione dei log (manuale, automatica o di archivio)

- I tuoi piani di emergenza nel caso di un oggetto danneggiato.

Se si sceglie un log circolare, il file system di log deve essere

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

Ciò consente al gestore code di scrivere in tutti i file primari e secondari. In circostanze eccezionali, il gestore code potrebbe scrivere un'estensione aggiuntiva oltre il numero di secondari. L'algoritmo precedente ne tiene conto.

Se si sceglie un log lineare, il file system del log deve essere significativamente più grande del log attivo.

Se si sceglie la gestione manuale dei log, il gestore code continua a scrivere su nuove estensioni di log quando ne ha bisogno ed è responsabilità dell'utente eliminarle (e archivarle) quando non sono più necessarie.

Quanto più grande deve essere il file system di log dipende in gran parte dalla strategia di eliminazione delle estensioni superflue o inattive.

È possibile decidere di archiviare ed eliminare le estensioni non appena diventano inattive (non necessarie per riavviare il ripristino) oppure è possibile decidere di archiviare ed eliminare solo le estensioni superflue (non necessarie per il supporto o riavviare il ripristino).

Se si stanno archiviando ed eliminando solo estensioni superflue, e se si dispone di un oggetto danneggiato, **MEDIALOG** non si sposterà in avanti, quindi nessun'altra estensione diventerà superflua. Si arresterà l'archiviazione e l'eliminazione delle estensioni fino a quando non si risolve il problema, forse ripristinando l'oggetto.

A meno che non si arresta il carico di lavoro, il tempo necessario per risolvere il problema dipende dalla dimensione del filesystem di log. Pertanto, è consigliabile avere un filesystem di log generoso quando si utilizza la registrazione lineare.

Se si sceglie un log lineare e la gestione dei log di archivio o automatica, il gestore code riutilizza le estensioni di log.

Le estensioni di log disponibili per essere riutilizzate hanno come prefisso la lettera R. Quando un'immagine del supporto viene registrata, poiché sono archiviate estensioni superflue, il gestore code può quindi riutilizzare tali estensioni.

Quindi, le estensioni di riutilizzo sono inferiori alla lunghezza dei dati scritti nel log tra le immagini del supporto:

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

Quando si registrano automaticamente le immagini del supporto magnetico e si imposta **IMGLOGLN**, `LogDataLengthBetweenMediaImages` può essere due volte **IMGLOGLN** perché **IMGLOGLN** non è un valore massimo fisso.

Quando si registrano manualmente le immagini del supporto o si registrano automaticamente per intervallo, `LogDataLengthBetweenMediaImages` dipende dal carico di lavoro e dall'intervallo tra le immagini.

Oltre alle estensioni attive e alle estensioni di riutilizzo, ci sono estensioni inattive (necessarie solo per il recupero dei supporti) e superflue (non necessarie per il riavvio o il ripristino dei supporti).

Quando si utilizza la gestione dei log di archivio o automatica, il gestore code non riutilizza le estensioni necessarie per il ripristino del supporto. Quindi, il numero di estensioni inattive dipende dalla frequenza con cui si stanno prendendo le immagini multimediali e se si stanno prendendo manualmente o automaticamente.

IMGINTVL e **IMGLOGLN** sono destinazioni, non un minimo o un massimo fisso tra le immagini del supporto. Tuttavia, quando si stima la dimensione massima del file system di log potrebbe essere necessario, è improbabile che le immagini di supporto automatiche vengano registrate più di due volte **IMGINTVL** o **IMGLOGLN** a parte.

Quando si ridimensiona il file system di log utilizzando la gestione automatica o di log di archivio, è necessario considerare anche cosa potrebbe accadere se una coda o un altro oggetto è danneggiato. In tal caso, il gestore code non è in grado di eseguire un'immagine del supporto dell'oggetto danneggiato e **MEDIALOG** non procederà.

Se il carico di lavoro continua, il log inattivo crescerà in modo illimitato poiché l'estensione più vecchia necessaria per il ripristino del supporto è ancora necessaria e non può essere riutilizzata. Se il carico di lavoro continua, si avrà fino a quando il file system del log non si riempie completamente per risolvere il problema, prima che il gestore code inizi a eseguire il rollback delle transazioni e potrebbe anche terminare bruscamente.

Quindi, per la gestione dei log di archiviazione e automatica:

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +  
(((TimeBetweenMediaImages *2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))  
* LogFilePages
```

Nota: L'algoritmo precedente presuppone che **SET LOG ARCHIVED** venga richiamato per ogni estensione, non appena non è più necessario per il ripristino del supporto, per la gestione del log di archiviazione.

Gestione dei log


Il prodotto supporta la gestione automatica dei log e il recupero automatico dei supporti dei log lineari. I log circolari sono quasi autogestiti, ma a volte necessitano di un intervento per risolvere i problemi di spazio.

Nota:  La gestione dei log di archiviazione e automatica non sono validi su IBM i.

Nella registrazione circolare, il gestore code recupera lo spazio liberato nei file di log. Questa attività non è evidente all'utente, e di solito non si vede la quantità di spazio su disco utilizzato ridurla, perché lo spazio allocato viene rapidamente riutilizzato.

È possibile eliminare i file secondari quando si utilizza la registrazione circolare. Per ulteriori informazioni, consultare [RESET QMGR TYPE \(REDUCELOG\)](#).

Nella registrazione lineare, il log potrebbe essere riempito se un punto di controllo non è stato utilizzato per un lungo periodo di tempo o se una transazione di lunga durata ha scritto un record di log molto tempo fa. Il gestore code tenta di eseguire i checkpoint abbastanza spesso per evitare il primo problema.

 Se il log si riempie, viene emesso il messaggio AMQ7463. Inoltre, se il log si riempie poiché una transazione di lunga durata ha impedito il rilascio dello spazio, viene emesso il messaggio AMQ7465.

Dei record di log, solo quelli scritti dall'inizio dell'ultimo punto di controllo completo e quelli scritti da qualsiasi transazione attiva, sono necessari per riavviare il gestore code.

Nel tempo, i record di log meno recenti scritti non sono più necessari per riavviare il gestore code.

Quando viene rilevata una transazione di lunga durata, l'attività viene pianificata per eseguire il rollback asincrono di tale transazione. Se, per qualche motivo imprevisto, il rollback asincrono ha avuto esito negativo, alcune chiamate MQI restituiscono MQRD_RESOURCE_PROBLEM in tale situazione.

Si noti che lo spazio è riservato per eseguire il commit o il rollback di tutte le transazioni in corso, quindi **MQCMIT** o **MQBACK** non dovrebbero avere esito negativo.

Un'applicazione per cui è stato eseguito il rollback di una transazione in questo modo non può eseguire operazioni **MQPUT** o **MQGET** successive specificando il punto di sincronizzazione nella stessa transazione.

Un tentativo di inserire o richiamare un messaggio nel punto di sincronizzazione in questo stato restituisce MQRD_BACKED_OUT. L'applicazione può quindi emettere **MQCMIT**, che restituisce MQRD_BACKED_OUT o **MQBACK** e avviare una nuova transazione. Quando è stato eseguito il rollback della transazione che utilizza troppo spazio di log, lo spazio di log viene rilasciato e il gestore code continua a funzionare normalmente.

Cosa succede quando un disco si riempie

Quando un gestore code è configurato per utilizzare la registrazione lineare, il componente di registrazione del gestore code reagisce a una condizione di disco pieno nei seguenti modi.

Se il disco che contiene i file di log diventa pieno:

- Il gestore code rileva questa condizione solo quando crea un nuovo file di log della dimensione richiesta, cosa che fa in anticipo rispetto a quando è necessario.
- Rileva la condizione di disco pieno quando il sistema operativo restituisce un errore dalla richiesta di estendere il file alla dimensione richiesta.
- Il gestore code emette il messaggio AMQ6708 al log degli errori del gestore code.
- Un record FFST (First Failure Support Technology) viene scritto nella directory degli errori a livello di sistema. Questo record fornisce i dettagli della condizione di disco pieno e deve essere conservato se è necessario contattare il supporto IBM .

I file di log vengono creati alla loro dimensione fissa, piuttosto che essere estesi quando i record di log vengono scritti in essi. Ciò significa che IBM MQ può esaurire lo spazio su disco solo quando sta creando un nuovo file; non può esaurire lo spazio quando scrive un record nel log. IBM MQ sa sempre quanto spazio è disponibile nei file di log esistenti e gestisce di conseguenza lo spazio all'interno dei file.

Quando si utilizza la registrazione lineare, è possibile utilizzare:

- Gestione automatica delle estensioni di log.

Consultare DISPLAY QMSTATUS per ulteriori informazioni sui nuovi attributi di log.

Inoltre, consultare i seguenti comandi o i relativi equivalenti PCF:

- Gestore code RESET
- SET LOG per piattaforme distribuite

- Le opzioni che controllano l'utilizzo delle immagini multimediali.

Consultare il comando ALTER QMGR e ALTER QUEUES per ulteriori informazioni su:

- IMGINTVL
- IMGLOGLN
- IMGRCOVO
- COCOVIMG
- IMGSCHEM

La registrazione circolare restituisce un problema di risorse.

Se lo spazio è ancora insufficiente, verificare che la configurazione del log nel file di configurazione del gestore code sia corretta. È possibile ridurre il numero di file di log primari o secondari in modo che il log non sia superiore allo spazio disponibile.

Non è possibile modificare la dimensione dei file di log per un gestore code esistente. Il gestore code richiede che tutte le estensioni log abbiano la stessa dimensione.

Gestione dei file di log

Assegnare spazio sufficiente per i file di log. Per la registrazione lineare, è possibile eliminare i vecchi file di log quando non sono più necessari.

Informazioni specifiche per la registrazione circolare

Se si utilizza la registrazione circolare, verificare che vi sia spazio sufficiente per conservare i file di log durante la configurazione del sistema (consultare “Stanza LogDefaults del file mq5.ini” a pagina 102 e “Stanza di log del file qm.ini” a pagina 141). La quantità di spazio su disco utilizzata dal log non aumenta oltre la dimensione configurata, incluso lo spazio per i file secondari da creare quando richiesto.

Informazioni specifiche per la registrazione lineare

Se si utilizza un log lineare, i file di log vengono aggiunti continuamente quando vengono registrati i dati e la quantità di spazio su disco utilizzato aumenta con il tempo. Se la velocità dei dati registrati è elevata, lo spazio su disco viene utilizzato rapidamente dai nuovi file di log.

Nel tempo, i file di log meno recenti per un log lineare non sono più richiesti per riavviare il gestore code o per eseguire il ripristino dei supporti di tutti gli oggetti danneggiati. I seguenti metodi determinano quali file di log sono ancora richiesti:

Messaggi di evento del programma di registrazione

Quando si verifica un evento significativo, ad esempio un'immagine del supporto di registrazione, vengono generati messaggi di evento del programma di registrazione. Il contenuto dei messaggi di evento del programma di registrazione specifica i file di log ancora richiesti per il riavvio del gestore code e per il recupero del supporto. Per ulteriori informazioni sui messaggi di evento del logger, consultare [Eventi del logger](#)

Stato gestore code

L'esecuzione del comando MQSC, DISPLAY QMSTATUS, o del comando PCF, Inquire Queue Manager Status, restituisce le informazioni sul gestore code, inclusi i dettagli dei file di log richiesti. Per ulteriori informazioni sui comandi MQSC, consultare [Amministrazione IBM MQ mediante i comandi MQSC](#) e per informazioni sui comandi PCF, consultare [Automazione delle attività di amministrazione](#).

Messaggi del gestore code

Periodicamente, il gestore code emette una coppia di messaggi per indicare quali file di log sono necessari:

- Il messaggio AMQ7467I fornisce il nome del file di log più vecchio richiesto per riavviare il gestore code. Questo file di log e tutti i file di log più recenti devono essere disponibili durante il riavvio del gestore code.
- Il messaggio AMQ7468I fornisce il nome del file di log più vecchio necessario per il ripristino del supporto.

Per determinare i file di log "più vecchi" e "più nuovi", utilizzare il numero del file di log piuttosto che le ore di modifica applicate dal file system.

Informazioni applicabili a entrambi i tipi di registrazione

Solo i file di log richiesti per il riavvio del gestore code, i file di log attivi, devono essere in linea. I file di log inattivi possono essere copiati su un supporto di archivio come ad esempio un nastro per il ripristino di emergenza e rimossi dalla directory di log. I file di log inattivi che non sono richiesti per il ripristino del supporto possono essere considerati come file di log superflui. È possibile eliminare i file di log superflui se non sono più di interesse per l'operazione.

Se non è possibile trovare alcun file di log necessario, viene emesso il messaggio dell'operatore AMQ6767E . Rendere il file di log e tutti i file di log successivi disponibili per il gestore code e ritentare l'operazione.

Ripulitura automatica delle aree di log - solo registrazione lineare



È possibile utilizzare la gestione automatica delle estensioni di log lineari non più necessarie per il recupero.

Si utilizza l'attributo **LogManagement** nella stanza Log del file qm.ini o utilizzando IBM MQ Explorer, per impostare la gestione automatica. Per ulteriori informazioni, consulta ["Stanza di log del file qm.ini"](#) a pagina 141.

Consultare il parametro [LOG](#) di **DISPLAY QMSTATUS** per ulteriori dettagli sull'operazione del log e i seguenti comandi per l'utilizzo del log:

- [Gestore code RESET](#)

- Impostazione log

Acquisizione automatica delle immagini multimediali - solo registrazione lineare

Esiste un interruttore generale per controllare se il gestore code scrive automaticamente le immagini dei supporti, il valore predefinito è che lo switch non è stato impostato.

È possibile controllare se si verifica l'imaging del supporto automatico e la frequenza del processo, utilizzando i seguenti attributi del gestore code:

IMGSCHED

Indica se il gestore code scrive automaticamente le immagini dei supporti

IMGINTVL

Frequenza per la scrittura di immagini multimediali, in minuti

IMGLOGLN

Megabyte di log scritti dalla precedente immagine del supporto di un oggetto.

Se si ha un momento critico durante il giorno quando il carico di lavoro è molto pesante e si desidera essere sicuri che la velocità di trasmissione del sistema non sia influenzata dall'utilizzo di immagini di supporto automatiche, è possibile disattivare temporaneamente l'imaging del supporto automatico impostando **IMGSCHED(MANUAL)**.

È possibile commutare **IMGSCHED** in qualsiasi momento durante il carico di lavoro.



Attenzione: MEDIALOG non viene spostato in avanti se non si stanno utilizzando immagini di supporto, quindi è necessario archiviare le estensioni o assicurarsi di disporre di spazio su disco sufficiente.

È anche possibile controllare le immagini dei supporti automatici e manuali per altri oggetti definiti dall'utente utilizzando l'attributo **IMGRCOVO** :

- Informazioni di autenticazione
- Canale
- Connessione client
- Listener
- Elenco nomi
- Processo
- Coda alias
- Coda locale
- Servizio
- Argomento

Per gli oggetti di sistema interni, ad esempio il catalogo oggetti e l'oggetto del gestore code, il gestore code scrive automaticamente le immagini dei supporti in base alle esigenze.

Consultare [ALTER QMGR](#) per ulteriori informazioni sugli attributi.

È inoltre possibile abilitare o disabilitare le immagini dei supporti automatici e manuali solo per le code dinamiche locali e permanenti. A tale scopo, utilizzare l'attributo coda **IMGRCOVQ** .

Consultare [ALTER QUEUES](#) per ulteriori informazioni sull'attributo **IMGRCOVQ** .

Note:

1. Le immagini dei supporti sono supportate solo se si utilizza la registrazione lineare. Se sono state abilitate le immagini di supporto automatiche, ma si sta utilizzando la registrazione circolare, viene emesso un messaggio di errore e l'attributo delle immagini di supporto automatico del gestore code è disabilitato.

2. Se sono state abilitate immagini di supporti automatici, ma non è stata specificata una frequenza, minuti o megabyte di log, viene emesso un messaggio di errore e non viene scritta alcuna immagine di supporti automatici.
3. È possibile registrare manualmente un'immagine di supporto utilizzando `rcdmqimg` una volta impostato **IMGSCHEM(AUTO)**, se lo si desidera.

Ciò consente di prendere immagini multimediali in un momento adatto per la propria azienda, ad esempio, quando il sistema è silenzioso. L'imaging del supporto automatico tiene conto di queste immagini del supporto manuale, poiché l'utilizzo di un'immagine del supporto manuale reimposta l'intervallo e la lunghezza del log, prima di cui viene presa la successiva immagine del supporto automatico.

4. Il gestore code scrive i messaggi persistenti solo nelle immagini dei supporti, non nei messaggi non persistenti. Ciò può ridurre la dimensione delle immagini del supporto durante la migrazione a versioni successive di IBM MQ .

Decidere come impostare **IMGLOGLN** e **IMGINTVL**

V 9.4.0 Per default, **IMGLOGLN** è impostata su `off` per i gestori code diversi dai gestori code della HA nativa. (I gestori code della HA nativo vengono creati con **IMGLOGLN** impostato sul valore del 25% dello spazio disponibile sul volume in cui devono essere scritti i log di ripristino.)

V 9.4.0 Per impostazione assunta, **IMGINTVL** è impostato su 60 minuti. L'intervallo specificato da **IMGINTVL** viene rispettato quando è stato eseguito un numero sufficiente di nuovi lavori sul gestore code per rendere utile la registrazione di una nuova immagine. Altrimenti l'acquisizione di nuove immagini viene ritardata.

È possibile modificare i valori di **IMGLOGLN** e **IMGINTVL** per ottenere la soluzione migliore per la propria configurazione. Rendere **IMGLOGLN** e **IMGINTVL** abbastanza grandi, in modo che il gestore code stia impiegando solo una frazione del suo tempo di registrazione delle immagini dei supporti, ma abbastanza piccoli in modo che:

- Gli oggetti danneggiati possono essere recuperati in un periodo di tempo ragionevole e
- Abbastanza piccolo in modo che il tuo log si adatti al tuo disco senza esaurire lo spazio.

Se si imposta **IMGLOGLN**, si consiglia di rendere **IMGLOGLN** molte volte la quantità di dati nelle code e molte volte la velocità dei dati del carico di lavoro. Maggiore è il valore di **IMGLOGLN**, minore è il tempo impiegato dal gestore code per la registrazione delle immagini dei supporti.

Allo stesso modo, se si imposta **IMGINTVL**, si consiglia di rendere **IMGINTVL** molte volte il tempo impiegato dal gestore code per registrare un'immagine del supporto. Puoi scoprire quanto tempo ci vuole per registrare un'immagine multimediale registrandone una manualmente.

Se si rendono **IMGLOGLN** e **IMGINTVL** troppo grandi, il recupero di un oggetto danneggiato potrebbe richiedere molto tempo, poiché tutte le estensioni dall'ultima immagine del supporto devono essere riprodotte.

Rendere **IMGLOGLN** e **IMGINTVL** sufficientemente piccoli, in modo che il tempo massimo impiegato per ripristinare un oggetto danneggiato sia accettabile.

Rendendo **IMGLOGLN** e **IMGINTVL** molto grandi, significa che il log cresce molto perché le immagini dei supporti vengono registrate raramente.



Attenzione: Assicurarsi che un log di questa dimensione si adatti comodamente al file system di log, poiché il carico di lavoro verrà ripristinato se il file system di log si riempie completamente.

È possibile impostare sia **IMGINTVL** che **IMGLOGLN**. Ciò può essere utile per garantire che le immagini dei supporti automatici vengano prese regolarmente durante un carico di lavoro pesante (controllato da **IMGLOGLN**), ma vengono ancora prese occasionalmente quando il carico di lavoro è molto leggero (controllato da **IMGINTVL**).

IMGINTVL e **IMGLOGLN** sono destinazioni per l'intervallo e la lunghezza dei dati di log tra cui vengono prese le immagini dei supporti automatici.

Questi attributi non devono essere considerati come un valore massimo o minimo fisso. Infatti, il gestore code potrebbe decidere di pianificare un'immagine del supporto automatico prima, se il gestore code percepisce che si tratta di un buon momento:

- Poiché la coda è vuota, prendere l'immagine del supporto è il più efficiente in termini di prestazioni e
- Un'immagine del supporto non è stata registrata per un certo periodo di tempo

A volte, il divario tra le immagini dei supporti automatici potrebbe essere un po' più lungo di **IMGINTVL** e **IMGLOGLN**, o di entrambi.

Il divario tra le immagini del supporto potrebbe essere maggiore di **IMGLOGLN** se la quantità di dati nelle code si sta avvicinando a **IMGLOGLN**. Il divario tra le immagini del supporto potrebbe essere maggiore di **IMGINTVL** se la registrazione di un'immagine del supporto richiede quasi il tempo **IMGINTVL**.

Questa non è una buona pratica perché il gestore code impiega la maggior parte del tempo a registrare immagini multimediali.

Quando si utilizza la registrazione automatica dell'immagine del supporto, il gestore code registra un'immagine del supporto per ogni oggetto e coda singolarmente, quindi il gestore code tiene traccia dell'intervallo e della lunghezza del log tra le immagini separatamente per ogni oggetto.

Gradualmente nel tempo, la registrazione di immagini multimediali diventa sfalsata, invece di registrare immagini multimediali per tutti gli oggetti allo stesso tempo. Questa sbalorditiva propaga l'impatto sulle prestazioni della registrazione delle immagini multimediali ed è un ulteriore vantaggio dell'utilizzo della registrazione automatica delle immagini multimediali rispetto alla registrazione manuale.

Acquisizione manuale di immagini multimediali - solo registrazione lineare

La registrazione di un'immagine di supporto di una coda implica la scrittura di tutti i messaggi persistenti da tale coda nel log. Per le code contenenti grandi volumi di dati di messaggi, ciò implica la scrittura di una grande quantità di dati nel log e questo processo può influire sulle prestazioni del sistema mentre sta accadendo.

È probabile che la registrazione delle immagini dei supporti di altri oggetti sia relativamente rapida, poiché l'immagine dei supporti di altri oggetti non contiene dati utente.

È necessario considerare attentamente quando registrare le immagini dei supporti delle code, in modo che il processo non interferisca con il carico di lavoro di picco.

È necessario registrare regolarmente l'immagine del supporto di tutti gli oggetti per aggiornare l'estensione del log meno recente necessaria per il ripristino del supporto.

Un buon momento per registrare l'immagine del supporto di una coda è quando è vuota, perché a quel punto non viene scritto alcun dato di messaggio nel log. Al contrario, un momento negativo è quando la coda è molto profonda o ha messaggi molto grandi su di essa.

Un buon momento per registrare l'immagine del supporto di una coda è quando il sistema è silenzioso; mentre un momento non corretto è durante il carico di lavoro di picco. Se il carico di lavoro è sempre silenzioso a mezzanotte, ad esempio, è possibile decidere di registrare le immagini multimediali a mezzanotte ogni notte.

Sfalsare la registrazione di ciascuna delle code può diffondere l'impatto sulle prestazioni e quindi ridurne l'effetto. Più tempo è stato dall'ultima registrazione delle immagini multimediali, più diventa importante registrarle, man mano che aumenta il numero di estensioni di log richieste per il ripristino dei supporti.

Nota: Quando si esegue il ripristino del supporto, tutti i file di log richiesti devono essere disponibili nella directory del file di log contemporaneamente. Assicurarsi di prendere regolarmente le immagini dei supporti di tutti gli oggetti che si desidera ripristinare per evitare di esaurire lo spazio su disco per contenere tutti i file di log richiesti.

Ad esempio, per ottenere un'immagine di supporto di tutti i tuoi oggetti nel tuo gestore code, esegui il comando **xcdmqimg** come mostrato nei seguenti esempi:

```
rcdmqimg -m QMNAME -t all *
```

```
rcdmqimg -m QMNAME -t all "*"
```

L'esecuzione di **rcdmqimg** sposta l'LSN (media log sequence number) in avanti. Per ulteriori dettagli sui numeri di sequenza log, consultare [“Dump del contenuto del log utilizzando il comando dmpmqlog”](#) a pagina 692. **rcdmqimg** non viene eseguito automaticamente, pertanto deve essere eseguito manualmente o da un'attività automatica creata. Per ulteriori informazioni relative a questo comando, consultare [rcdmqimg](#) e [dmpmqlog](#).

La registrazione manuale delle immagini dei supporti con **rcdmqimg** per gestire lo spazio di log non è richiesta, se si è scelto di utilizzare la registrazione lineare con l'imaging dei supporti automatico controllato dal gestore code.

Nota: I messaggi AMQ7467 e AMQ7468 possono essere emessi anche quando si esegue il comando **rcdmqimg**.

Immagini di supporti parziali

Si consiglia di utilizzare i messaggi IBM MQ solo per i dati che si prevede vengano utilizzati nel prossimo futuro, in modo che ogni messaggio si trovi in una coda per un periodo di tempo relativamente breve.

Al contrario, non è consigliabile utilizzare i messaggi IBM MQ per memorizzare i dati a lungo termine come un database.

È anche buona norma assicurarsi che le code siano relativamente basse e che non sia buona pratica avere code profonde i cui messaggi sono stati in coda per un lungo periodo di tempo.

Seguendo queste linee guida, si abilita il gestore code per ottimizzare le prestazioni della registrazione automatica delle immagini multimediali.

La registrazione dell'immagine del supporto di una coda vuota è molto efficiente (dal punto di vista delle prestazioni), mentre prendere l'immagine del supporto di una coda con una grande quantità di dati su di essa è molto inefficiente, perché tutti questi dati devono essere scritti nel log nell'immagine del supporto.

Per le code poco profonde con messaggi immessi di recente su di esso, il gestore code può effettuare un'ulteriore ottimizzazione.

Se tutti i messaggi attualmente presenti nella coda sono stati inseriti nel passato recente, il gestore code potrebbe essere in grado di registrare l'immagine del supporto per un determinato periodo di tempo (*punto di recupero*) prima dell'inserimento di tutti i messaggi e quindi di registrare l'immagine della coda vuota. Questo processo è molto basso in termini di prestazioni.

Se tutti i messaggi che si trovavano nella coda al punto di ripristino sono stati successivamente ricevuti, non è necessario registrare tali messaggi nell'immagine del supporto, poiché non si trovano più nella coda.

Questa viene definita *immagine di supporto parziale*. Quindi, nel caso improbabile in cui la coda debba essere ripristinata, tutti i record di log relativi a questa coda dall'ultima immagine del supporto verranno riprodotti, in modo da ripristinare tutti i messaggi immessi di recente.

Anche se ci fossero alcuni messaggi nella coda al punto di ripristino, che sono attualmente nella coda (e quindi devono essere registrati nell'immagine del supporto parziale), è ancora più efficiente registrare questa immagine del supporto parziale più piccola, piuttosto che un'immagine del supporto completo di tutti i messaggi.

Garantire che i messaggi rimangano in coda per un breve periodo di tempo potrebbe migliorare le prestazioni della registrazione automatica delle immagini multimediali.

Ubicazione file di log

Quando si sceglie un percorso per i propri file di log, tenere presente che l'operazione è gravemente compromessa se IBM MQ non riesce a formattare un nuovo log a causa della mancanza di spazio su disco.

Se si utilizza un log circolare, verificare che vi sia spazio sufficiente sull'unità per almeno i file di log primari configurati. Inoltre, lasciare spazio per almeno un file di log secondario, che è necessario se il log deve crescere.

Se si utilizza un log lineare, consentire una quantità di spazio notevolmente maggiore; lo spazio utilizzato dal log aumenta continuamente man mano che vengono registrati i dati.

È necessario posizionare i file di log su un'unità disco separata dai dati del gestore code.

L'integrità dei dati su questo dispositivo è fondamentale - dovresti consentire la ridondanza integrata.

Potrebbe anche essere possibile posizionare i file di log su più unità disco in una disposizione di mirroring. Ciò protegge da malfunzionamenti dell'unità che contiene il log. Senza il mirroring, si potrebbe essere costretti a tornare all'ultimo backup del proprio sistema IBM MQ.

ALW Modifica del log del gestore code da lineare a circolare

È possibile modificare il tipo di log del gestore code da lineare a circolare utilizzando il comando **migmqlog**.

Prima di iniziare

Decidere se si desidera utilizzare la registrazione lineare o circolare esaminando [Tipi di registrazione](#).

Decidere se si desidera modificare il tipo di log o spostare il log in una nuova ubicazione. Quando si sposta il log in una nuova posizione utilizzando il comando **migmqlog**, il percorso del log nel file `qm.ini` viene aggiornato in modo che quando si avvia il gestore code utilizzi il log modificato. È possibile specificare una nuova ubicazione utilizzando l'opzione **-ld**. Se si sta spostando il gestore code da un vecchio disco a un nuovo disco Advanced Format, potrebbe essere utile utilizzare l'opzione **-ld**.

Procedura

1. Accedere come membro del gruppo `mqm`.
2. Assicurarsi di disporre di spazio sufficiente per modificare il log. È necessario assicurarsi che vi sia spazio per almeno i file di log primari configurati e per un file di log secondario.
3. Se non è stato ancora fatto, arrestare il gestore code utilizzando il comando **endmqm -w**.
4. Se non è stato ancora fatto, eseguire un backup del gestore code.

Per ulteriori informazioni, consultare “Backup dei dati del gestore code” a pagina 696e [“Backup dei file di configurazione dopo la creazione di un gestore code”](#) a pagina 14.

5. Eseguire il comando **migmqlog**:

- Se si sceglie di modificare il tipo di log senza modificare l'ubicazione del log, utilizzare il seguente comando:

```
migmqlog -m QMgrName -lc
```

- Se si sceglie di modificare il tipo di log e spostare il log in una nuova ubicazione, utilizzare il seguente comando:

```
migmqlog -m QMgrName -lc -ld NewLogLocation
```

dove *NewLogLocation* è un percorso file assoluto che specifica la nuova ubicazione per il file di log. Non utilizzare un percorso file relativo con il parametro **-ld**.

Per ulteriori informazioni, consultare [migmqlog](#).

Se, per qualsiasi motivo, ad esempio a causa di un'interruzione dell'alimentazione, il comando **migmqlog** si arresta prima di aver completato il processo, eseguire nuovamente lo stesso comando **migmqlog** sui log parzialmente modificati per completare le modifiche.

Risultati

Il comando viene eseguito e il tipo di log del gestore code viene aggiornato. Tenere presente che il completamento di **migmqlog** potrebbe richiedere alcuni minuti se il log è molto grande. Tuttavia, il comando emette messaggi di avanzamento di volta in volta.

Attività correlate

[“Modifica del log del gestore code da circolare a lineare” a pagina 685](#)

È possibile modificare il tipo di log del gestore code da circolare a lineare utilizzando il comando **migmqlog**.

 *Modifica del log del gestore code da circolare a lineare*

È possibile modificare il tipo di log del gestore code da circolare a lineare utilizzando il comando **migmqlog**.

Prima di iniziare

Decidere se si desidera utilizzare la registrazione lineare o circolare esaminando [Tipi di registrazione](#).

Decidere se si desidera modificare il tipo di log o spostare il log in una nuova ubicazione. Quando si sposta il log in una nuova posizione utilizzando il comando **migmqlog**, il percorso del log nel file `qm.ini` viene aggiornato in modo che quando si avvia il gestore code utilizzi il log modificato. È possibile specificare una nuova ubicazione utilizzando l'opzione **-ld**. Se si sta spostando il gestore code da un vecchio disco a un nuovo disco Advanced Format, potrebbe essere utile utilizzare l'opzione **-ld**.

Informazioni su questa attività



Attenzione: Dopo aver modificato il log, un'immagine del supporto non sarà stata registrata all'avvio del gestore code. Pianificare il modo in cui si intende registrare le immagini multimediali, impostando automaticamente gli attributi:

- IMGSCHEd
- IMGINTVL
- IMGLOGLN
- IMGRCOVO
- COCOVIMG

in [ALTER QMGRo](#) manualmente eseguendo periodicamente **rcdmqimg**.

Procedura

1. Accedere come membro del gruppo `mqm`.
2. Assicurarsi di disporre di spazio sufficiente per modificare il log. Lo spazio utilizzato da un log lineare aumenta continuamente man mano che vengono registrati i dati.
3. Se non è stato ancora fatto, arrestare il gestore code utilizzando il comando **endmqm -w**.
4. Se non è stato ancora fatto, eseguire un backup del gestore code.
Per ulteriori informazioni, consultare [“Backup dei dati del gestore code” a pagina 696](#) e [“Backup dei file di configurazione dopo la creazione di un gestore code” a pagina 14](#).
5. Eseguire il comando **migmqlog**. Tenere presente che il completamento di **migmqlog** potrebbe richiedere alcuni minuti se il log è molto grande. Tuttavia, il comando emette messaggi di avanzamento di volta in volta.

- Se si sceglie di modificare il tipo di log senza modificare l'ubicazione del log, utilizzare il seguente comando:

```
migmqlog -m QMgrName -ll
```

- Se si sceglie di modificare il tipo di log e spostare il log in una nuova ubicazione, utilizzare il seguente comando:

```
migmqlog -m QMgrName -ll -ld NewLogLocation
```

dove *NewLogLocation* è un percorso file assoluto che specifica la nuova ubicazione per il file di log. Non utilizzare un percorso file relativo con il parametro **-ld**.

Per ulteriori informazioni, consultare [migmqlog](#).

Se, per qualsiasi motivo, ad esempio a causa di un'interruzione dell'alimentazione, il comando **migmqlog** si arresta prima di aver completato il processo, eseguire nuovamente lo stesso comando **migmqlog** sui log parzialmente modificati per completare le modifiche.

6. Avviare il gestore code e impostare il ripristino dell'immagine e gli attributi della coda appropriati per il proprio ambiente.
7. Considerare quando registrare le immagini manuali per gli oggetti che sono recuperabili.

Attività correlate

“Modifica del log del gestore code da lineare a circolare” a pagina 684

È possibile modificare il tipo di log del gestore code da lineare a circolare utilizzando il comando **migmqlog**.

Determinazione dei file di log superflui - solo registrazione lineare

Per la registrazione circolare, non eliminare mai i dati dalla directory di log. Quando si gestiscono i file di log lineari, è importante verificare quali file possono essere eliminati o archiviati. Queste informazioni ti aiuteranno a prendere questa decisione.

Non utilizzare le ore di modifica del file system per determinare i file di log "più vecchi". Utilizzare solo il numero del file di log. L'utilizzo dei file di log da parte del gestore code segue regole complesse, inclusa la pre - assegnazione e la formattazione dei file di log prima che siano necessari. È possibile visualizzare i file di log con le ore di modifica che potrebbero essere fuorvianti se si tenta di utilizzare queste ore per determinare l'età relativa.

Per determinare il file di log più vecchio necessario, sono disponibili tre posti da utilizzare:

- Comando DISPLAY QMSTATUS
- Messaggi di evento del programma di registrazione e, infine
- Messaggi di log degli errori

Per il comando DISPLAY QMSTATUS, per stabilire l'estensione del log meno recente necessaria per:

- Riavviare il gestore code, immettere il comando DISPLAY QMSTATUS RECLOG.
- Eseguire il ripristino del supporto, immettere il comando DISPLAY QMSTATUS MEDIALOG.
- Determinare il nome per la notifica di archiviazione, immettere il comando DISPLAY QMSTATUS ARCHLOG.

È possibile ridurre il numero di estensioni di log secondarie quando si utilizza la registrazione circolare immettendo il comando **RESET QMGR TYPE (REDUCELOG)**.

In generale, un numero di file di log inferiore implica un log meno recente. A meno che non si abbia un turnover del file di log molto elevato, dell'ordine di 3000 file di log al giorno per 10 anni, non è necessario soddisfare il numero di wrapping di 9 999 999. In questo caso, è possibile archiviare qualsiasi file di log con un numero inferiore al valore RECLOG ed è possibile eliminare qualsiasi file di log con un numero inferiore ai valori RECLOG e MEDIALOG.



Attenzione: Il file di log viene riportato a capo, quindi il numero successivo dopo 9 999 999 è zero.

Coldstart: Cosa fare se le estensioni di log sono mancanti o corrotte

Se l'azienda perde alcune o tutte le estensioni di log necessarie per riavviare il ripristino, il gestore code non sarà in grado di riprodurre il log di recupero e quindi non sarà in grado di riavviarlo. Se si richiede il riavvio del gestore code quando il log di ripristino è danneggiato in qualsiasi modo, a discapito del mantenimento dell'integrità dei dati, è possibile farlo, anche se fortemente sconsigliato. Questo processo è noto come *avvio a freddo* di un gestore code.

Importante: L'avvio a freddo di un gestore code deve essere considerato solo in circostanze eccezionali e comporta rischi di integrità dei dati come descritto in questa pagina. IBM suggerisce di ricreare un gestore code, preferendo l'avvio a freddo, in risposta a file di dati danneggiati.

Se è richiesto un avvio a freddo per motivi operativi, contattare il rappresentante del supporto IBM per esaminare la causa principale del problema. È necessario sostituire un gestore code avviato a freddo con un gestore code ricostruito quanto prima.

Gli effetti di coldstart

Su coldstart, il gestore code crea un log di recupero vuoto e si basa sui dati presenti nei file di coda e in altri file di oggetto nello stato esistente. Poiché i dati nei file della coda possono essere incongruenti, i messaggi potrebbero essere persi, duplicati, danneggiati o incongruenti.

Il gestore code memorizza la configurazione di tutti gli altri oggetti persistenti nel log di ripristino e nei file di oggetti. Anche altri dati di stato interni vengono registrati nel log di ripristino, quindi su coldstart, i dati di stato interni vengono reimpostati e tutti gli altri dati di configurazione potrebbero non essere accurati.

Gli effetti di coldstart sono imprevedibili e di ampia portata, quindi si dovrebbe evitare un coldstart a meno che non sia assolutamente necessario. Dopo l'avvio a freddo, le informazioni nei file della coda e dell'oggetto possono essere così incongruenti che il gestore code non verrà riavviato.

Se il gestore code non viene riavviato, non esiste un modo semplice per rilevare i dati del messaggio o la configurazione su cui è possibile fare affidamento e gli elementi che non è possibile utilizzare. Inoltre, dopo un avvio a freddo, le code potrebbero essere danneggiate e quindi diventare completamente inutilizzabili.

Inoltre, se è possibile ottenere da, o inserire in, una particolare coda, i messaggi su di essa potrebbero essere danneggiati, mancanti o duplicati. Le transazioni e i canali potrebbero essere bloccati in dubbio. Anche se il coldstart del gestore code viene eseguito correttamente e le code sembrano intatte, gli effetti imprevedibili del coldstart potrebbero non essere realizzati fino a molto tempo dopo.

Cosa fare se è necessario eseguire il coldstart

L'esecuzione di un avvio a freddo non deve essere considerata una pratica operativa standard e IBM sconsiglia vivamente di eseguire questa operazione. Tuttavia, se si è in una posizione in cui è assolutamente necessario avviare a freddo un gestore code, contattare [Supporto IBM MQ](#).

Il processo di avvio a freddo di un gestore code era molto più complicato per un gestore code lineare rispetto a un gestore code circolare. In IBM MQ 9.1.3, il processo coldstart è stato molto semplificato e non implica più la copia o la ridenominazione delle estensioni di log.

Contattare il supporto IBM, che fornirà una chiave che verrà passata al comando **strmqm** per avviare a freddo un gestore code.



Attenzione: Il comando coldstart comporta ancora gli stessi rischi di perdita dell'integrità dei dati di un coldstart manuale e IBM sconsiglia vivamente di eseguire questa operazione.

Eliminare le future partenze a freddo: una richiesta

Il comando strmqm richiede una chiave per eseguire il comando coldstart, poiché IBM MQ desidera che l'utente contatti il supporto IBM MQ se è necessario eseguire il comando coldstart, poiché IBM MQ è interessato a comprendere come si è arrivati a questa situazione.

Chiaramente coldstart è qualcosa che è meglio evitare. IBM MQ ha compiuto notevoli sforzi per assicurarsi che non sia necessario eseguire il coldstart del proprio gestore code e IBM desidera scoprire se il prodotto può fare di più per ridurre il coldstart.

Precauzioni per evitare un avvio a freddo

Il metodo di registrazione predefinito quando si crea un gestore code è la registrazione circolare. Con la registrazione circolare si consente al gestore code un numero particolare di estensioni di log primario e

secondario di una determinata dimensione. Creare il file system di log abbastanza grande da contenere tutte le estensioni di log primario e secondario e non è mai necessario gestirle.

In alternativa, è possibile utilizzare la registrazione lineare rispetto a quella circolare. La registrazione lineare consente di recuperare code e altri oggetti, nell'improbabile caso in cui vengano danneggiati. Ma per impostazione predefinita, la registrazione lineare richiede di eliminare le estensioni di log che non sono più necessarie per il riavvio o il recupero del supporto. Si fa riferimento a questo come alla gestione del log manuale.

Quando si gestiscono le estensioni di log in questo modo, è possibile eliminare inavvertitamente troppe estensioni di log e quindi finire per dover eseguire il comando `coldstart`. Per ridurre questo rischio, utilizzare la gestione log automatica, in modo che il gestore code gestisca le estensioni log per conto dell'utente.

La procedura ottimale consiste nell'inserire il log di ripristino in un file system di log separato che contiene solo il log di ripristino. Se si inserisce il log di ripristino nello stesso filesystem del resto del gestore code, è possibile che il filesystem si riempia accidentalmente, forse a causa di file di coda di grandi dimensioni. Rendere la directory di log per il gestore code un file system separato oppure specificare un file system di log differente utilizzando l'opzione della riga comandi **-ld** nel comando **crtmqm**.

Se il filesystem che contiene i file della coda si riempie, potrebbe non essere possibile inserire tali code, ma il gestore code continua l'esecuzione. Se il file system che contiene il log di ripristino si riempie, il gestore code termina bruscamente e non si riavvierà fino a quando non si libera spazio.

Fare attenzione a non eliminare le estensioni di log necessarie per riavviare il ripristino, altrimenti potrebbe essere necessario eseguire il comando `coldstart`. A volte si potrebbe scoprire che è necessario eseguire il `coldstart` perché il disco non è riuscito che contiene il relativo log di ripristino. La procedura ottimale è quella di inserire il log di ripristino su un disco replicato e quindi ridurre il rischio di un arresto anomalo del disco.

Lo spostamento dei messaggi e della configurazione in un nuovo gestore code di sostituzione evita la possibilità di problemi in corso con un gestore code che è stato precedentemente avviato a freddo.

Tenere presente quali gestori code sono stati precedentemente avviati a freddo, anche se sono stati avviati a freddo molto tempo fa e sono stati arrestati, riavviati e migrati nel frattempo. Quando si contatta il supporto IBM, indicare se il gestore code è stato precedentemente avviato a freddo e, in tal caso, fornire quante più informazioni possibili su ciò che ha causato il requisito di un avvio a freddo.

Utilizzo del log per il ripristino

Puoi utilizzare le informazioni dai log per aiutarti a recuperare da errori.

Esistono diversi modi in cui i dati possono essere danneggiati. IBM MQ ti aiuta a recuperare da:

- Un oggetto dati danneggiato
- Una perdita di potenza nel sistema
- Un errore di comunicazione

Questa sezione esamina come vengono utilizzati i log per risolvere questi problemi.

Ripristino da perdita di alimentazione o errori di comunicazione

IBM MQ può eseguire il ripristino da errori di comunicazioni e perdita di alimentazione. A volte può anche recuperare da altri tipi di problemi, come l'eliminazione involontaria di un file.

In caso di errore di comunicazione, i messaggi persistenti rimangono nelle code fino a quando non vengono rimossi da un'applicazione ricevente. Se il messaggio è in fase di trasmissione, rimane nella coda di trasmissione fino a quando non può essere trasmesso correttamente. Per eseguire il ripristino da un errore di comunicazioni, di solito è possibile riavviare i canali utilizzando il collegamento non riuscito.

Se si perde l'alimentazione, quando il gestore code viene riavviato IBM MQ ripristina le code allo stato di cui è stato eseguito il commit al momento dell'errore. Ciò garantisce che nessun messaggio persistente venga perso. I messaggi non persistenti vengono eliminati; non sopravvivono quando IBM MQ si arresta bruscamente.

Recupero degli oggetti danneggiati

Esistono modi in cui un oggetto IBM MQ può diventare inutilizzabile, ad esempio a causa di danni involontari. È quindi necessario ripristinare il proprio sistema completo o una parte di esso. L'azione richiesta dipende dal momento in cui viene rilevato il danno, se il metodo di registrazione selezionato supporta il ripristino del supporto e quali oggetti sono danneggiati.

Ripristino supporti

È possibile registrare le immagini dei supporti per gli oggetti in modo che possano essere ripristinati in caso di danneggiamento. Questa funzione è disponibile solo per i gestori code che utilizzano la registrazione lineare o la registrazione replicata e, per la registrazione lineare, solo per gli oggetti definiti come recuperabili. Si definisce che i tipi di oggetto sono recuperabili utilizzando gli attributi del gestore code **IMGRCOVO** e **IMGRCOVQ**, consultare [ALTER QMGR](#). Se un oggetto non definito come recuperabile è danneggiato, le opzioni di recupero sono le stesse della registrazione circolare.

Il ripristino dei supporti ricrea gli oggetti dalle informazioni registrate in un log lineare o in un log replicato. Ad esempio, se un file oggetto viene inavvertitamente cancellato o diventa inutilizzabile per qualche altro motivo, il ripristino del supporto può ricrearlo. Le informazioni nel log richieste per il ripristino del supporto di un oggetto sono denominate *immagine supporto*.

Un'immagine del supporto è una sequenza di record di log che contengono un'immagine di un oggetto da cui l'oggetto stesso può essere ricreato.

Il primo record di log richiesto per ricreare un oggetto è noto come *record di ripristino del supporto*; è l'inizio dell'ultima immagine del supporto per l'oggetto. Il record di recupero supporti di ciascun oggetto è una delle informazioni registrate durante un punto di controllo.

Quando un oggetto viene ricreato dalla sua immagine multimediale, è anche necessario riprodurre tutti i record di log che descrivono gli aggiornamenti eseguiti sull'oggetto dall'ultima immagine.

Considerare, ad esempio, una coda locale che ha un'immagine dell'oggetto della coda presa prima che un messaggio persistente venga inserito nella coda. Per ricreare l'immagine più recente dell'oggetto, è necessario riprodurre le voci del log che registrano l'inserimento del messaggio nella coda, oltre a riprodurre l'immagine stessa.

Quando un oggetto viene creato, i record di log scritti contengono informazioni sufficienti per ricrearlo completamente. Questi record costituiscono la prima immagine multimediale dell'oggetto. Quindi, ad ogni arresto, il gestore code registra automaticamente le immagini dei media come segue:

- Immagini di tutte le code e gli oggetti del processo non locali
- Immagini di code locali vuote

Le immagini del supporto possono anche essere registrate manualmente utilizzando il comando **rcdmqing**, descritto nella sezione [rcdmqing](#). Questo comando scrive un'immagine del supporto dell'oggetto IBM MQ.

Il gestore code registra automaticamente le immagini dei supporti se è impostato **IMGSCHED(AUTO)**. Per ulteriori informazioni, consultare [ALTER QMGR](#) per informazioni su **IMGINTVL** e **INGLOGLN**.

Quando è stata scritta un'immagine del supporto, solo i log che contengono l'immagine del supporto e tutti i log creati dopo questo periodo di tempo sono richiesti per ricreare gli oggetti danneggiati. Il vantaggio della creazione di immagini multimediali dipende da fattori quali la quantità di memoria libera disponibile e la velocità con cui vengono creati i file di log.

Ripristino da immagini multimediali

Un gestore code recupera automaticamente alcuni oggetti dall'immagine del supporto durante l'avvio del gestore code. Recupera automaticamente una coda se è stata coinvolta in una transazione che era incompleta quando il gestore code è stato chiuso l'ultima volta e risulta danneggiata o danneggiata durante il processo di riavvio.

È necessario recuperare altri oggetti manualmente, utilizzando il comando **rcrmqobj**, che riproduce i record nel log per creare nuovamente l'oggetto IBM MQ. L'oggetto viene ricreato dalla sua ultima

immagine trovata nel log, insieme a tutti gli eventi di log applicabili tra l'ora in cui l'immagine è stata salvata e l'ora in cui è stato emesso il comando di ricreazione. Se un oggetto IBM MQ viene danneggiato, le uniche azioni valide che possono essere eseguite sono l'eliminazione o la ricreazione mediante questo metodo. I messaggi non persistenti non possono essere recuperati in questo modo.

Per ulteriori informazioni sul comando **rcrmqobj**, consultare [rcrmqobj](#).

Il file di log contenente il record di ripristino del supporto e tutti i successivi file di log devono essere disponibili nella directory del file di log quando si tenta il ripristino del supporto di un oggetto. Se non è possibile trovare un file richiesto, viene emesso il messaggio operatore AMQ6767 e l'operazione di ripristino del supporto ha esito negativo. Se non si prendono immagini di supporti regolari degli oggetti che si desidera ricreare, lo spazio su disco potrebbe non essere sufficiente per contenere tutti i file di log richiesti per ricreare un oggetto.

OpenShift **V 9.4.0** I gestori code della HA nativa utilizzano la registrazione replicata. Tali gestori code tentano il ripristino automatico degli oggetti idonei quando viene rilevato un danno. Una volta avviati, i gestori code della HA nativa, per impostazione predefinita, tentano automaticamente il ripristino asincrono quando viene rilevato un danno all'oggetto. Il ripristino potrebbe non essere immediatamente possibile se, ad esempio, l'oggetto è utilizzato da un'applicazione oppure se le estensioni di log richieste per il ripristino del supporto non sono disponibili. In queste situazioni, l'elaborazione del recupero asincrono ritenta periodicamente. Se il problema che ha impedito il ripristino viene risolto, l'oggetto verrà ripristinato al successivo tentativo oppure l'oggetto può essere ripristinato manualmente, utilizzando il comando **rcrmqobj**.

Quali file oggetto esistono

Il gestore code memorizza gli attributi degli oggetti definiti in **runmqsc** in file su disco. Questi file di oggetti si trovano nelle sottodirectory nella directory di dati del gestore code.

Linux **AIX** Ad esempio, su piattaforme AIX and Linux, i canali sono memorizzati in `/var/mqm/qmgrs/qmgr/channel`.

I dati in questi file oggetto sono l'immagine supporto degli oggetti. Se questi file oggetto vengono eliminati o danneggiati, l'oggetto memorizzato in quel file è danneggiato. Utilizzando un gestore code di registrazione lineare, gli oggetti danneggiati possono essere recuperati dal log utilizzando il comando [rcrmqobj](#). I gestori code della registrazione replicata (HA nativa) tentano automaticamente di recuperare gli oggetti danneggiati quando vengono rilevati.

La maggior parte dei file oggetto contiene solo gli attributi dell'oggetto, quindi i file canale contengono gli attributi dei canali. Le eccezioni sono:

- Catalogo

Il catalogo di oggetti cataloga tutti gli oggetti di tutti i tipi ed è memorizzato in `qmanager/QMQMOBJCAT`.

- File di sincronizzazione

Il file di sincronizzazione contiene dati di stato interni associati a tutti i canali.

- Code

I file di coda contengono sia i messaggi su quella coda che gli attributi di quella coda.

Notare che non vi è alcun oggetto di catalogo o syncfile esposto in **runmqsc** o IBM MQ Explorer.

Il catalogo e il gestore code possono essere registrati, ma non recuperati. Se questi oggetti vengono danneggiati, il gestore code termina preventivamente e questi oggetti vengono ripristinati automaticamente al riavvio.

Le sottoscrizioni non vengono elencate negli oggetti da registrare o ripristinare, poiché le sottoscrizioni durevoli vengono memorizzate su una coda di sistema. Per registrare o ripristinare sottoscrizioni durevoli, registrare o ripristinare `SYSTEM.DURABLE.SUBSCRIBER.QUEUE`.

Ripristino degli oggetti danneggiati durante l'avvio

Se il gestore code rileva un oggetto danneggiato durante l'avvio, l'azione che intraprende dipende dal tipo di oggetto e se il gestore code è configurato per supportare il ripristino dei supporti.

Se l'oggetto gestore code è danneggiato, il gestore code non può essere avviato a meno che non sia in grado di ripristinare l'oggetto. Se il gestore code è configurato con un log lineare e quindi supporta il ripristino dei supporti, IBM MQ tenta automaticamente di ricreare l'oggetto gestore code dalle relative immagini dei supporti. Se il metodo di log selezionato non supporta il ripristino del supporto, è possibile ripristinare un backup del gestore code o eliminare il gestore code.

Se le transazioni erano attive quando il gestore code è stato arrestato, le code locali che contengono i messaggi persistenti, senza commit immessi o ricevuti all'interno di queste transazioni sono richieste anche per avviare correttamente il gestore code. Se una di queste code locali risulta danneggiata e il gestore code supporta il recupero dei supporti, tenta automaticamente di ricrearle dalle relative immagini dei supporti. Se una qualsiasi delle code non può essere ripristinata, IBM MQ non può essere avviato.

Se le code locali danneggiate contenenti messaggi non sottoposti a commit vengono rilevate durante l'elaborazione di avvio su un gestore code che non supporta il ripristino dei supporti, le code vengono contrassegnate come oggetti danneggiati e i messaggi non sottoposti a commit vengono ignorati. Questa situazione è dovuta al fatto che non è possibile eseguire il ripristino dei supporti degli oggetti danneggiati su un gestore code di questo tipo e l'unica azione rimasta è eliminarli. Viene emesso il messaggio AMQ7472 per segnalare eventuali danni.

Recupero di oggetti danneggiati in altri momenti

Il ripristino dei supporti degli oggetti è automatico solo durante l'avvio (diverso dai gestori code della HA nativa, che utilizzano il ripristino automatico per impostazione predefinita). In altri casi, quando viene rilevato un danneggiamento dell'oggetto, viene emesso il messaggio dell'operatore AMQ7472 e la maggior parte delle operazioni che utilizzano l'oggetto non riesce con il codice di ritorno MQRC_OBJECT_DAMAGED. Se l'oggetto del gestore code è danneggiato in qualsiasi momento dopo l'avvio del gestore code, il gestore code esegue una chiusura preventiva. Quando un oggetto è stato danneggiato, è possibile eliminarlo oppure, se il gestore code sta utilizzando un log lineare, tentare di ripristinarlo dall'immagine del supporto utilizzando il comando `rcrmqobj` (per ulteriori dettagli, consultare [rcrmqobj](#)).


Se una coda (o un altro oggetto) viene danneggiato, **MEDIALOG** non si sposterà in avanti. Questo perché **MEDIALOG** è l'estensione meno recente richiesta per il ripristino del supporto. Se il tuo carico di lavoro continua, **CURRLOG** continuerà ad andare avanti e quindi verranno scritte nuove estensioni. A seconda della tua configurazione (inclusa la tua impostazione **LogManagement**), questo potrebbe iniziare a riempire il tuo file system di log. Se il file system di log si riempie completamente, viene eseguito il rollback delle transazioni e il gestore code potrebbe terminare bruscamente. Quindi, quando una coda viene danneggiata, si potrebbe avere solo una quantità limitata di tempo per agire prima che il gestore code termini. La quantità di tempo a disposizione dipende dalla frequenza con cui il carico di lavoro sta causando la scrittura di nuove estensioni da parte del gestore code e dalla quantità di spazio libero disponibile nel file system del log.

Se si utilizza la gestione manuale del log, è possibile che si stiano archiviando estensioni non necessarie per il ripristino del riavvio e quindi eliminarle dal file system del log, anche se sono ancora necessarie per il recupero del supporto. Ciò è accettabile fino a quando è possibile ripristinarle dall'archivio quando necessario. Questa politica non determina il riempimento del file system di log quando una coda viene danneggiata e **MEDIALOG** smette di procedere. Tuttavia, se si archiviano ed eliminano solo le estensioni che non sono necessarie per il riavvio o il ripristino del supporto, il file system del log inizia a riempirsi se una coda viene danneggiata.

Se si utilizza la gestione dei log di archiviazione o automatica, il gestore code non riutilizzerà le estensioni ancora necessarie per il recupero del supporto, anche se è possibile che siano state archiviate e che abbiano inviato una notifica al gestore code utilizzando [SET LOG ARCHIVED](#). Di conseguenza, se una coda viene danneggiata, il file system di log inizierà a riempirsi.

Se una coda viene danneggiata, verranno scritti gli FFDC OBJECT DAMAGED e **MEDIALOG** smetterà di procedere. L'oggetto danneggiato può essere identificato da FFDC o perché è l'oggetto con il più vecchio **MEDIALOG** quando si visualizza il suo stato in **runmqsc**.

Se il file system di log è pieno e si è preoccupati che il carico di lavoro venga ripristinato perché il file system di log sta diventando pieno, il ripristino dell'oggetto o la sospensione del carico di lavoro potrebbe impedire che ciò accada.

 Nel caso di gestori code HA nativi, che utilizzano la registrazione replicata, viene tentato il recupero automatico degli oggetti danneggiati. Una volta avviati, i gestori code della HA nativa, per impostazione predefinita, tentano automaticamente il ripristino asincrono quando viene rilevato un danno all'oggetto. Il ripristino potrebbe non essere immediatamente possibile se, ad esempio, l'oggetto è utilizzato da un'applicazione oppure se le estensioni di log richieste per il ripristino del supporto non sono disponibili. In queste situazioni, l'elaborazione del recupero asincrona ritenta periodicamente. Se il problema che ha impedito il ripristino viene risolto, l'oggetto verrà ripristinato al successivo tentativo oppure l'oggetto può essere ripristinato manualmente, utilizzando il comando **rcrmqobj**.

Protezione dei file di log IBM MQ

Non toccare i file di log quando un gestore code è in esecuzione, il ripristino potrebbe essere impossibile. Utilizzare l'autorizzazione superutente o mqm per proteggere i file di log da modifiche involontarie.

Non rimuovere manualmente i file di log attivi quando è in esecuzione un gestore code IBM MQ. Se un utente elimina inavvertitamente i file di log che un gestore code deve riavviare, IBM MQ **non** emette errori e continua a elaborare i dati *inclusi i messaggi persistenti*. Il gestore code viene arrestato normalmente, ma il riavvio potrebbe non riuscire. Il recupero dei messaggi diventa quindi impossibile.

Gli utenti che dispongono dell'autorizzazione per rimuovere i log utilizzati da un gestore code attivo hanno anche l'autorizzazione per eliminare altre importanti risorse del gestore code (come i file di coda, il catalogo oggetti e i file eseguibili IBM MQ). Possono quindi danneggiare, forse a causa dell'inesperienza, un gestore code in esecuzione o inattivo in un modo rispetto al quale IBM MQ non può proteggersi.

Prestare attenzione quando si conferiscono le autorizzazioni di super utente o mqm.

Dump del contenuto del log utilizzando il comando dmpmqlog

Come utilizzare il comando `dmpmqlog` per eseguire il dump del contenuto del log del gestore code.

Utilizzare il comando `dmpmqlog` per eseguire il dump del contenuto del log del gestore code. Per impostazione predefinita, viene eseguito il dump di tutti i record di log attivi, ossia, il comando avvia il dump dall'inizio del log (di solito l'avvio dell'ultimo punto di controllo completato).

Il log può essere scaricato solo quando il gestore code non è in esecuzione. Poiché il gestore code prende un punto di controllo durante l'arresto, la porzione attiva del log di solito contiene un numero ridotto di record di log. Tuttavia, è possibile utilizzare il comando `dmpmqlog` per eseguire il dump di più record di log utilizzando una delle opzioni riportate di seguito per modificare la posizione iniziale del dump:

- Avviare il dump dalla *base* del log. La base del log è il primo record di log nel file di log che contiene l'intestazione del log. La quantità di dati aggiuntivi di cui viene eseguito il dump in questo caso dipende da dove è posizionata l'intestazione del log nel file di log. Se è vicino all'inizio del file di log, viene eseguito il dump solo di una piccola quantità di dati aggiuntivi. Se l'intestazione è vicina alla fine del file di log, viene eseguito il dump di un numero significativamente maggiore di dati.
- Specificare la posizione iniziale del dump come singolo record di log. Ogni record di log è identificato da un *LSN (log sequence number)* univoco. Nel caso di registrazione circolare, questo record di log iniziale non può essere prima della base del log; questa restrizione non si applica ai log lineari. Potrebbe essere necessario ripristinare i file di log inattivi prima di eseguire il comando. È necessario specificare un LSN valido, ricavato dall'output `dmpmqlog` precedente, come posizione iniziale.

Ad esempio, con la registrazione lineare è possibile specificare `nextlsn` dall'ultimo output `dmpmqlog`. `nextlsn` viene visualizzato in `Log File Header` e indica l'LSN del successivo record di log da

scrivere. Utilizzarlo come posizione iniziale per formattare tutti i record di registrazione scritti dall'ultima volta che è stato eseguito il dump della registrazione.

- **Solo per i log lineari**, è possibile indicare a `dmpmqlog` di avviare la formattazione dei record di log da una determinata estensione del file di log. In tal caso, `dmpmqlog` prevede di trovare questo file di log e ogni file successivo nella stessa directory dei file di log attivi. Questa opzione non si applica ai log circolari, dove `dmpmqlog` non può accedere ai record di log prima della base del log.

L'output del comando `dmpmqlog` è `Log File Header` e una serie di record di log formattati. Il gestore code utilizza diversi record di log per registrare modifiche ai propri dati.

Alcune informazioni formattate vengono utilizzate solo internamente. Il seguente elenco include i record di log più utili:

Intestazione file di log

Ogni log ha una singola intestazione del file di log, che è sempre la prima cosa formattata dal comando `dmpmqlog`. Contiene i seguenti campi:

<i>logattivo</i>	Il numero di estensioni log principali.
<i>loginactive</i>	Il numero di estensioni log secondarie.
<i>dimensione log</i>	Il numero di pagine da 4 KB per estensione.
<i>base</i>	Il primo LSN nell'estensione log contenente l'intestazione del log.
<i>Nextlsn</i>	L'LSN del successivo record di log da scrivere.
<i>intestazioni</i>	L'LSN del record di log all'inizio del log.
<i>tailsn</i>	L'LSN che identifica la posizione di coda del registro.
<i>hflag1</i>	Se il log è CIRCULAR o LOG RETAIN (lineare).
<i>HeadExtentHeadExtent</i>	L'estensione del log contenente l'intestazione del log.

Intestazione record log

Ogni record di log all'interno del log ha un'intestazione fissa che contiene le seguenti informazioni:

<i>LSN</i>	Il numero di sequenza log.
<i>LogRecdLogRecd</i>	Il tipo di record di log.
<i>XTranid</i>	L'identificativo della transazione associato a questo record di log (se presente). Un <i>TranType</i> di MQI indica una transazione solo IBM MQ. Un <i>TranType</i> di XA è coinvolto con altri gestori risorse. Gli aggiornamenti coinvolti nella stessa unità di lavoro hanno lo stesso <i>XTranid</i> .
<i>QueueName</i>	La coda associata a questo record di log (se presente).
<i>QID</i>	L'identificativo interno univoco per la coda.
<i>PrevLSN</i>	L'LSN del record di log precedente all'interno della stessa transazione (se presente).

Avvia gestore code

Questo log indica che il gestore code è stato avviato.

<i>StartDate</i>	La data di avvio del gestore code.
<i>StartTime</i>	L'ora in cui è stato avviato il gestore code.

Arresta gestore code

Questo log indica che il gestore code è stato arrestato.

<i>StopDate</i>	La data in cui il gestore code è stato arrestato.
<i>StopTime</i>	L'ora in cui il gestore code è stato arrestato.
<i>ForceFlag</i>	Il tipo di arresto utilizzato.

Avvia punto di controllo

Indica l'inizio di un checkpoint del gestore code.

Termina punto di controllo

Indica la fine di un checkpoint del gestore code.

<i>ChkPtLSN</i>	L'LSN del record di log che ha avviato questo punto di controllo.
-----------------	---

Inserisci messaggio

Registra un messaggio persistente inserito in una coda. Se il messaggio è stato inserito nel punto di sincronizzazione, l'intestazione del record di log contiene un *XTranId* non null. Il resto del record contiene:

<i>MapIndex</i>	Un identificativo per il messaggio sulla coda. Può essere utilizzato per corrispondere al MQGET corrispondente utilizzato per ottenere questo messaggio dalla coda. In questo caso, è possibile trovare un record di log <i>Get Message</i> successivo contenente gli stessi <i>QueueName</i> e <i>MapIndex</i> . A questo punto l'identificativo <i>MapIndex</i> può essere riutilizzato per un messaggio di inserimento successivo a tale coda.
<i>Dati</i>	Nel dump esadecimale per questo record di log sono contenuti vari dati interni, seguiti da una rappresentazione del descrittore del messaggio (eyecatcher MD) e quindi i dati del messaggio stessi.

Inserisci parte

I messaggi permanenti troppo grandi per un singolo record di log vengono registrati come più record di log *Put Part* seguiti da un singolo record *Put Message*. Se sono presenti *Put Part* record, il campo *PrevLSN* concatenerà i record *Put Part* e il record finale *Put Message*.

<i>Dati</i>	Continua i dati del messaggio in cui è stato lasciato il record di log precedente.
-------------	--

Richiama messaggio

Vengono registrati solo i richiami dei messaggi persistenti. Se il messaggio è stato ottenuto nel punto di sincronizzazione, l'intestazione del record di log contiene un *XTranId* non null. Il resto del record contiene:

<i>MapIndex</i>	Identifica il messaggio che è stato richiamato dalla coda. Il record di log <i>Put Message</i> più recente contenente lo stesso <i>QueueName</i> e <i>MapIndex</i> identifica il messaggio che è stato richiamato.
<i>QPriorità</i>	La priorità del messaggio richiamato dalla coda.

Avvia transazione

Indica l'inizio di una nuova transazione. Un *TranType* di MQI indica una transazione solo IBM MQ. Un *TranType* di XA indica uno che coinvolge altri gestori risorse. Tutti gli aggiornamenti effettuati da questa transazione avranno lo stesso *XTranId*.

Prepara transazione

Indica che il gestore code è preparato per eseguire il commit degli aggiornamenti associati al *XTranId* specificato. Questo record di log viene scritto come parte di un commit a due fasi che coinvolge altri gestori risorse.

Esegui commit transazione

Indica che il gestore code ha eseguito il commit di tutti gli aggiornamenti effettuati da una transazione.

Transazione di rollback

Ciò denota l'intenzione del gestore code di eseguire il rollback di una transazione.

Fine transazione

Indica la fine di una transazione di cui è stato eseguito il rollback.

Tabella transazioni

Questo record viene scritto durante il punto di sincronizzazione. Registra lo stato di ogni transazione che ha effettuato aggiornamenti permanenti. Per ogni transazione vengono registrate le seguenti informazioni:

<i>XTranid</i>	L'identificativo della transazione.
<i>FirstLSN</i>	L'LSN del primo record di log associato alla transazione.
<i>LastLSN</i>	L'LSN dell'ultimo record di log associato alla transazione.

Partecipanti alla transazione

Questo record di log viene scritto dal componente Gestore transazioni XA del gestore code. Registra i gestori risorse esterni che partecipano alle transazioni. Per ogni partecipante viene registrato quanto segue:

<i>Nome RM</i>	Il nome del gestore risorse.
<i>IDRM</i>	L'identificativo del gestore risorse. Viene registrato anche nei successivi record di log <i>Transaction Prepared</i> che registrano le transazioni globali a cui partecipa il gestore risorse.
<i>SwitchFile</i>	Il file di caricamento switch per questo gestore risorse.
<i>XAOpenString</i>	La stringa di apertura XA per questo gestore risorse.
<i>XACloseString</i>	La stringa di chiusura XA per questo gestore risorse.

Transazione preparata

Questo record di log viene scritto dal componente Gestore transazioni XA del gestore code. Indica che la transazione globale specificata è stata preparata correttamente. A ciascun gestore risorse partecipante verrà richiesto di eseguire il commit. Il *RMID* di ogni gestore risorse preparato viene registrato nel record di log. Se il gestore code stesso partecipa alla transazione, sarà presente un *Participant Entry* con un *RMID* uguale a zero.

Transazione non utilizzata

Questo record di log viene scritto dal componente Gestore transazioni XA del gestore code. Segue il record di log *Transaction Prepared* quando la decisione di commit è stata consegnata a ogni partecipante.

Elimina coda

Questo registra il fatto che tutti i messaggi in una coda sono stati eliminati, ad esempio, utilizzando il comando MQSC CLEAR QUEUE.

Attributi Coda

Registra l'inizializzazione o la modifica degli attributi di una coda.

Crea oggetto

Registra la creazione di un oggetto IBM MQ .

<i>ObjName</i>	Il nome dell'oggetto creato.
<i>UserId</i>	L'ID utente che esegue la creazione.

Elimina oggetto

Questo registra l'eliminazione di un oggetto IBM MQ .

<i>ObjName</i>	Il nome dell'oggetto che è stato eliminato.
----------------	---

Backup e ripristino dei dati del gestore code IBM MQ

È possibile proteggere i gestori code da possibili danneggiamenti causati da errori hardware eseguendo il backup dei gestori code e dei dati del gestore code, eseguendo solo il backup della configurazione del gestore code e utilizzando un gestore code di backup.

Informazioni su questa attività



Avvertenza: È necessario prestare molta attenzione se si sposta un gestore code su un sistema operativo diverso. Per ulteriori informazioni, consultare [Spostamento di un gestore code su un sistema operativo diverso](#).

Periodicamente, è possibile adottare misure per proteggere i gestori code da possibili danneggiamenti causati da malfunzionamenti hardware. Esistono tre modi per proteggere un gestore code:

Eseguire il backup dei dati del gestore code

Se l'hardware ha esito negativo, è possibile che venga forzato l'arresto di un gestore code. Se i dati di log del gestore code vengono persi a causa di un errore hardware, il gestore code potrebbe non essere in grado di riavviarsi. Se si esegue il backup dei dati del gestore code, è possibile ripristinare alcuni o tutti i dati del gestore code persi.

In generale, più spesso si esegue il backup dei dati del gestore code, meno dati si perdono in caso di errore hardware che causa la perdita di integrità del log di recupero.

Per eseguire il backup dei dati del gestore code, il gestore code non deve essere in esecuzione.

Solo backup della configurazione del gestore code

Se l'hardware ha esito negativo, è possibile che venga forzato l'arresto di un gestore code. Se la configurazione del gestore code e i dati di log vengono persi a causa di un errore hardware, il gestore code non è in grado di riavviarsi o di essere ripristinato dal log. Se si esegue il backup della configurazione del gestore code, è possibile ricreare il gestore code e tutti i relativi oggetti dalle definizioni salvate.

Per eseguire il backup della configurazione del gestore code, il gestore code deve essere in esecuzione.

Utilizza un gestore code di backup

Se l'errore hardware è grave, un gestore code potrebbe non essere recuperabile. In questa situazione, se il gestore code non recuperabile ha un gestore code di backup dedicato, il gestore code di backup può essere attivato al posto del gestore code non recuperabile. Se viene aggiornato regolarmente, il log del gestore code di backup può contenere i dati di log che includono l'ultimo log completo dal gestore code non recuperabile.

Un gestore code di backup può essere aggiornato mentre il gestore code esistente è in esecuzione.

Procedura

- Per eseguire il backup e ripristinare i dati del gestore code, consultare:
 - [“Backup dei dati del gestore code”](#) a pagina 696.
 - [“Ripristino dei dati del gestore code”](#) a pagina 697.
- Per eseguire il backup e il ripristino della configurazione del gestore code, consultare:
 - [“Backup della configurazione del gestore code”](#) a pagina 698
 - [“Ripristino della configurazione del gestore code”](#) a pagina 699
- Per creare, aggiornare e avviare un gestore code di backup, consultare [“Utilizzo di un gestore code di backup”](#) a pagina 700.

Backup dei dati del gestore code

Il backup dei dati del gestore code consente di evitare possibili perdite di dati causate da errori hardware.

Prima di iniziare

Prima di avviare il backup del gestore code, verificare che il gestore code non sia in esecuzione. Se si tenta di eseguire un backup di un gestore code in esecuzione, il backup potrebbe non essere congruente a causa degli aggiornamenti in corso durante la copia dei file. Se possibile, arrestare il gestore code eseguendo il comando **endmqm -w** (un arresto di attesa), solo se non riesce, utilizzare il comando **endmqm -i** (un arresto immediato).

Informazioni su questa attività

Per eseguire una copia di backup dei dati di un gestore code, completare le seguenti attività:

Procedura

1. Ricercare le directory in cui il gestore code inserisce i suoi dati e i suoi file di log utilizzando le informazioni nei file di configurazione.

Per ulteriori informazioni, consultare [“Modifica delle informazioni di configurazione IBM MQ nei file .ini su Multiplatforms”](#) a pagina 93.

Nota: I nomi visualizzati nella directory vengono trasformati per garantire che siano compatibili con la piattaforma su cui si utilizza IBM MQ. Per ulteriori informazioni sulle trasformazioni dei nomi, consultare [Informazioni sui nomi file di IBM MQ](#).


2. Eseguire copie di tutte le directory dei dati e dei file di log del gestore code, incluse tutte le sottodirectory.

Assicurarsi di non perdere alcun file, specialmente il file di controllo log, come descritto in [“Aspetto dei log”](#) a pagina 664, e i file di configurazione come descritto in [“File di inizializzazione e di configurazione”](#) a pagina 254. Alcune delle directory potrebbero essere vuote, ma sono necessarie tutte per ripristinare il backup in un secondo momento.

Per la registrazione circolare, eseguire il backup dei dati del gestore code e delle directory dei file di log contemporaneamente in modo da ripristinare una serie coerente di dati e log del gestore code.

Per la registrazione lineare, eseguire contemporaneamente il backup dei dati del gestore code e delle directory dei file di log. È possibile ripristinare solo i file di dati del gestore code se è disponibile una sequenza completa corrispondente di file di log.

3. Conservare le proprietà dei file.

 Per i sistemi IBM MQ for UNIX e Linux, è possibile eseguire questa operazione con il comando **tar**. (Se si dispone di code più grandi di 2 GB, non è possibile utilizzare il comando **tar**. Per ulteriori informazioni, consultare [Abilitazione di code di grandi dimensioni](#).)

Nota: Quando si esegue l'aggiornamento a IBM WebSphere MQ 7.5 e versioni successive, assicurarsi di eseguire un backup del file `qm.ini` e delle voci del Registro di sistema. Le informazioni sul gestore code sono memorizzate nel file `qm.ini` e possono essere utilizzate per ripristinare una versione precedente di IBM MQ.

Attività correlate

[Arresto di un gestore code](#)

[“Backup dei file di configurazione dopo la creazione di un gestore code”](#) a pagina 14

Le informazioni di configurazione IBM MQ sono memorizzate nei file di configurazione su AIX, Linux, and Windows. Dopo aver creato un gestore code, eseguire il backup dei propri file di configurazione. Quindi, se si crea un altro gestore code che causa problemi, è possibile ripristinare i backup una volta rimossa l'origine del problema.

Ripristino dei dati del gestore code

Effettuare le operazioni riportate di seguito per ripristinare un backup dei dati di un gestore code.

Prima di iniziare

Prima di iniziare il backup, verificare che il gestore code non sia in esecuzione.

Quando si ripristina un backup di un gestore code in un cluster, consultare “Ripristino di un gestore code cluster” a pagina 382 e [Cluster: disponibilità, più istanze e ripristino di emergenza](#) per ulteriori informazioni.

Nota: Quando si esegue l'aggiornamento a una versione successiva di IBM MQ, assicurarsi di eseguire un backup del file **.ini** e delle voci di registro. Le informazioni sul gestore code sono memorizzate nel file **.ini** e possono essere utilizzate per ripristinare una versione precedente di IBM MQ.

Procedura

1. Individuare le directory in cui il gestore code inserisce i suoi dati e i suoi file di log, utilizzando le informazioni nei file di configurazione.
2. Svuotare le directory in cui si desidera inserire i dati di cui è stato eseguito il backup.
3. Copiare i dati del gestore code di cui è stato eseguito il backup e i file di log nelle ubicazioni corrette. Accertarsi di disporre di un file di controllo log e dei file di log.

Per la registrazione circolare, eseguire il backup dei dati del gestore code e delle directory dei file di log contemporaneamente in modo da ripristinare una serie coerente di dati e log del gestore code.

Per la registrazione lineare, eseguire contemporaneamente il backup dei dati del gestore code e delle directory dei file di log. È possibile ripristinare solo i file di dati del gestore code se è disponibile una sequenza completa corrispondente di file di log.

4. Aggiornare i file di informazioni di configurazione.
Verificare che i IBM MQ e i file di configurazione del gestore code siano congruenti in modo che IBM MQ possa ricercare i dati ripristinati nelle ubicazioni corrette.
5. Controllare la struttura di directory risultante per assicurarsi di disporre di tutte le directory richieste.
Per ulteriori informazioni sulle directory e le sottodirectory IBM MQ, consultare [Directory structure on Windows systems](#) e [Directory content on AIX and Linux systems](#).

Risultati

Se è stato eseguito il backup e il ripristino corretto dei dati, il gestore code verrà avviato.

Multi

Backup della configurazione del gestore code

Il backup della configurazione del gestore code può essere utile per ricreare un gestore code dalle relative definizioni se la configurazione del gestore code e i dati di log vengono persi a causa dell'errore hardware e il gestore code non è in grado di riavviare o di essere ripristinato dal log.

Informazioni su questa attività

ALW

Su AIX, Linux, and Windows, puoi utilizzare il comando **dmpmqcfg** per eseguire il dump della configurazione di un gestore code IBM MQ.

IBM i

In IBM i, è possibile utilizzare il comando Dump di MQ Configuration (**DMPMQCFG**) per eseguire il dump degli oggetti di configurazione e delle autorizzazioni per un gestore code.

Procedura

1. Assicurarsi che il programma di gestione code sia in esecuzione.
2. A seconda della piattaforma, utilizzare uno dei comandi riportati di seguito per eseguire il back up della configurazione del gestore code:

- ▶ ALW Su AIX, Linux, and Windows: eseguire il comando Dump MQ Configuration, **dmpmqcfcg** utilizzando l'opzione di formattazione predefinita (- f mqsc) MQSC e tutti gli attributi (-a), utilizzare il reindirizzamento dell'output standard per memorizzare le definizioni in un file. Ad esempio:

```
dmpmqcfcg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

- ▶ IBM i Su IBM i: eseguire il comando di configurazione Dump MQ (**DMPMQMCFG**) utilizzando l'opzione di formattazione predefinita OUTPUT (*MQSC) e EXPATTR (*ALL), utilizzare TOFILE e TOMBR per memorizzare le definizioni in un membro del file fisico. Ad esempio:

```
DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC)
TOMBR(MYQMGRDEF)
```

Attività correlate

“Ripristino della configurazione del gestore code” a pagina 699

È possibile ripristinare la configurazione per un gestore code da un backup verificando prima che il gestore code sia in esecuzione e quindi eseguendo il comando appropriato per la propria piattaforma.

Riferimenti correlati

[dmpmqcfcg \(configurazione gestore code dump\)](#)

[Dump della configurazione di MQ \(DMP MQMCFG\)](#)

▶ Multi Ripristino della configurazione del gestore code

È possibile ripristinare la configurazione per un gestore code da un backup verificando prima che il gestore code sia in esecuzione e quindi eseguendo il comando appropriato per la propria piattaforma.

Informazioni su questa attività

▶ ALW Su AIX, Linux, and Windows, è possibile utilizzare il comando **runmqsc** per ripristinare la configurazione di un gestore code IBM MQ .

▶ IBM i Su IBM i, è possibile utilizzare il comando **STRMQMMQSC** per ripristinare le autorizzazioni e gli oggetti di configurazione per un gestore code.

Procedura

1. Assicurarsi che il programma di gestione code sia in esecuzione.

Tenere presente che, se i danni ai dati e ai log non sono recuperabili in altri modi, è possibile che il gestore code sia stato ricreato.

2. A seconda della piattaforma, utilizzare uno dei comandi seguenti per ripristinare la configurazione del gestore code:

- ▶ ALW Su AIX, Linux, and Windows, eseguire **runmqsc** rispetto al gestore code, utilizzare il reindirizzamento di immissione standard per ripristinare le definizioni da uno script generato dal comando Dump di MQ Configuration (**dmpmqcfcg**) (consultare [“Backup della configurazione del gestore code” a pagina 698](#)). Ad esempio:

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

- ▶ IBM i Su IBM i: eseguire **STRMQMMQSC** sul gestore code e utilizzare i parametri **SRCMBR** e **SRCFILE** per ripristinare le definizioni dal membro del file fisico generato dal comando Dump di

configurazione MQ (**DMPMQMCFG**) (vedere [“Backup della configurazione del gestore code”](#) a pagina 698). Ad esempio:

```
STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
```

Attività correlate

[“Backup della configurazione del gestore code”](#) a pagina 698

Il backup della configurazione del gestore code può essere utile per ricreare un gestore code dalle relative definizioni se la configurazione del gestore code e i dati di log vengono persi a causa dell'errore hardware e il gestore code non è in grado di riavviare o di essere ripristinato dal log.

Riferimenti correlati

[dmpmqcfg](#) (configurazione gestore code dump)

[runmqsc](#) (esecuzione comandi MQSC)

[Dump della configurazione di MQ \(DMP MQMCFG\)](#)

[Comandi di avvio IBM MQ \(STRMQMMQSC\)](#)

Utilizzo di un gestore code di backup

Un gestore code esistente può avere un gestore code di backup dedicato per scopi di ripristino di emergenza.

Informazioni su questa attività

Un gestore code di backup è una copia inattiva del gestore code esistente. Se il gestore code esistente diventa irreversibile a causa di un grave errore hardware, il gestore code di backup può essere portato in linea per sostituire il gestore code irreversibile.

I file di log del gestore code esistenti devono essere regolarmente copiati nel gestore code di backup per garantire che il gestore code di backup rimanga un metodo efficace per il ripristino di emergenza. Il gestore code esistente non deve essere arrestato per i file di log da copiare, tuttavia è necessario copiare un file di log solo se il gestore code ha terminato la scrittura su di esso; consultare [“Aggiornamento di un gestore code di backup”](#) a pagina 701 per informazioni su come assicurarsi che uno specifico file di log non venga più scritto, in modo che possa essere copiato in modo sicuro.

Nota: Poiché il log del gestore code esistente viene continuamente aggiornato, esiste sempre una leggera discrepanza tra il log del gestore code esistente e i dati del log copiati nel log del gestore code di backup. Gli aggiornamenti regolari al gestore code di backup riducono la discrepanza tra i due log.

Se un gestore code di backup deve essere portato in linea, deve essere attivato e quindi avviato. Il requisito di attivare un gestore code di backup prima che venga avviato è una misura preventiva da proteggere dall'avvio accidentale di un gestore code di backup. Una volta attivato, un gestore code di backup non può essere più aggiornato.

Importante: Una volta che il vecchio gestore code di backup è diventato il nuovo gestore code attivo, per qualsiasi motivo, non esiste più un gestore code di backup. Si tratta effettivamente di una forma di replica asincrona, per cui si prevede che il nuovo gestore code attivo sia logicamente in ritardo rispetto al vecchio gestore code attivo. Pertanto, il vecchio gestore code attivo non funge più da backup per il nuovo gestore code attivo.

Procedura

- Per informazioni sull'utilizzo di un gestore code di backup, consultare i seguenti argomenti:
 - [“Creazione di un gestore code di backup”](#) a pagina 701
 - [“Aggiornamento di un gestore code di backup”](#) a pagina 701
 - [“Avvio di un gestore code di backup”](#) a pagina 702

Concetti correlati

[“Registrazione: verifica che i messaggi non vengano persi” a pagina 664](#)

IBM MQ registra tutte le modifiche significative ai dati permanenti controllati dal gestore code in un log di ripristino.

Creazione di un gestore code di backup

Creare un gestore code di backup come una copia inattiva del gestore code esistente.

Informazioni su questa attività

Importante: È possibile utilizzare un gestore code di backup solo quando si utilizza la registrazione lineare.

Un gestore code di backup richiede quanto segue:

- Per avere gli stessi attributi del gestore code esistente, ad esempio il nome del gestore code, il tipo di registrazione e la dimensione del file di log.
- Essere sulla stessa piattaforma del gestore code esistente.
- Essere a un livello di codice uguale o superiore a quello del gestore code esistente.

Procedura

1. Creare un gestore code di backup per il gestore code esistente utilizzando il comando di controllo **crtmqm**.
2. Eseguire copie di tutte le directory dei file di log e dei dati del gestore code esistenti, incluse tutte le sottodirectory, come descritto in [“Backup dei dati del gestore code” a pagina 696](#).
3. Sovrascrivere le directory dei file di log e dei dati del gestore code di backup, incluse le sottodirectory, con le copie prese dal gestore code esistente.
4. Eseguire il comando di controllo **strmqm** sul gestore code di backup come mostrato nel seguente esempio:

```
strmqm -x BackupQMName
```

Questo comando contrassegna il gestore code come gestore code di backup all'interno di IBM MQe riproduce tutte le estensioni di log copiate per portare il gestore code di backup al passo con il gestore code esistente.

Riferimenti correlati

[crtmqm \(crea gestore code\)](#)

[strmqm \(avvio gestore code\)](#)

Aggiornamento di un gestore code di backup

Per garantire che un gestore code di backup rimanga un metodo efficace per il ripristino di emergenza, è necessario aggiornarlo regolarmente.

Informazioni su questa attività

L'aggiornamento regolare riduce la discrepanza tra il log del gestore code di backup e il log del gestore code corrente. Non è necessario arrestare il gestore code prima di eseguire il backup.



Avvertenza: Se si copia una serie di log non contigui nella directory di log del gestore code di backup, vengono riprodotti solo i log fino al punto in cui viene trovato il primo log mancante.

Procedura

1. Immettere il seguente comando di script (MQSC) sul gestore code di cui eseguire il backup:

```
RESET QMGR TYPE(ADVANCELOG)
```

Questa operazione arresta qualsiasi scrittura nel log corrente e quindi avanza la registrazione del gestore code all'estensione di log successiva. Ciò garantisce il backup di tutte le informazioni registrate all'ora corrente.

2. Ottenere il numero (nuovo) di estensione del log attivo corrente emettendo il seguente comando Script (MQSC) sul gestore code di cui eseguire il backup:

```
DIS QMSTATUS CURRLOG
```

3. Copiare i file di estensione log aggiornati dalla directory di log del gestore code corrente alla directory di log del gestore code di backup.

Copiare tutte le estensioni di log dall'ultimo aggiornamento e fino (ma non incluso) all'estensione corrente indicata in "2" a pagina 702. Copiare solo i file di estensione log, quelli che iniziano con "S. ...".

4. Eseguire il comando di controllo **strmqm** sul gestore code di backup come mostrato nel seguente esempio:

```
strmqm -r BackupQMName
```

Ciò riproduce tutte le estensioni di log copiate e porta il gestore code di backup al passo con il gestore code. Al termine della riproduzione, si riceve un messaggio che identifica tutte le estensioni di log richieste per il ripristino del riavvio e tutte le estensioni di log richieste per il ripristino del supporto.

Riferimenti correlati

[RESET QMGR](#)

[VISUALIZZAZIONE QMSTATUS](#)

[strmqm \(avvio gestore code\)](#)

Avvio di un gestore code di backup

È possibile sostituire un gestore code di backup con un gestore code non recuperabile.

Informazioni su questa attività

Quando si ripristina un backup di un gestore code in un cluster, consultare "Ripristino di un gestore code cluster" a pagina 382 e [Cluster: disponibilità, più istanze e ripristino di emergenza](#) per ulteriori informazioni.

Se un gestore code non recuperabile ha un gestore code di backup dedicato, è possibile attivare il gestore code di backup al posto del gestore code non recuperabile.

Quando un gestore code non recuperabile viene sostituito con un gestore code di backup, è possibile che alcuni dei dati del gestore code non recuperabili vengano persi. La quantità di dati persi dipende dall'ultimo aggiornamento del gestore code di backup. Più recentemente l'ultimo aggiornamento, meno perdita di dati del gestore code.

Nota: Anche se i file di log e i dati del gestore code si trovano in directory differenti, assicurarsi di eseguire il backup e ripristinare le directory contemporaneamente. Se i dati e i file di log del gestore code hanno età diverse, il gestore code non è in uno stato valido e probabilmente non verrà avviato. Anche se inizia, è probabile che i tuoi dati siano corrotti.

Procedura

1. Eseguire il comando di controllo **strmqm** per attivare il gestore code di backup come mostrato nel seguente esempio:

```
strmqm -a BackupQMName
```

Il gestore code di backup è attivato. Ora che è attivo, il gestore code di backup non può più essere aggiornato.

2. Eseguire il comando di controllo **strmqm** per avviare il gestore code di backup come mostrato nel seguente esempio:

strmqm BackupQMName

IBM MQ considera questo come un ripristino di riavvio e utilizza il log dal gestore code di backup. Durante l'ultimo aggiornamento al gestore code di backup, si verificherà la ripetizione, pertanto verrà eseguito il rollback solo delle transazioni attive dall'ultimo punto di controllo registrato.

3. Riavviare tutti i canali.
4. Controllare la struttura di directory risultante per assicurarsi di disporre di tutte le directory richieste. Per ulteriori informazioni sulle directory e le sottodirectory IBM MQ , consultare [Pianificazione del supporto del file system](#).
5. Accertarsi di disporre di un file di controllo log e dei file di log. Verificare inoltre che IBM MQ e i file di configurazione del gestore code siano congruenti in modo che IBM MQ possa ricercare i dati ripristinati nelle posizioni corrette.

Risultati

Se il backup e il ripristino dei dati sono stati eseguiti correttamente, il gestore code viene avviato.

Attività correlate

[“Riavvio dei canali arrestati” a pagina 246](#)

Quando un canale passa allo stato ARRESTATO, è necessario riavviare manualmente il canale.

Riferimenti correlati

[strmqm \(avvio gestore code\)](#)

Multi

Modifiche al ripristino da errori cluster sui server su Multiplatforms

Il gestore code riesegue le operazioni che hanno causato problemi, fino a quando i problemi non vengono risolti. Se, dopo cinque giorni, i problemi non vengono risolti, il gestore code si arresta per evitare che la cache diventi più obsoleta.

Il gestore code riesegue le operazioni che hanno causato problemi, fino a quando i problemi non vengono risolti. Se, dopo cinque giorni, i problemi non vengono risolti, il gestore code si arresta per evitare che la cache diventi più obsoleta. Man mano che la cache diventa più obsoleta, causa un numero maggiore di problemi.

Ogni aspetto della gestione del cluster viene gestito per un gestore code dal processo del gestore repository locale, amqzmf. Il processo viene eseguito su tutti i gestori code, anche se non sono presenti definizioni cluster.

IBM MQ, invece di arrestare il gestore repository e continuare senza di esso, il gestore repository riesegue le operazioni non riuscite. Se il gestore code rileva un problema con il gestore repository, segue una delle due linee di azione.

1. Se l'errore non compromette l'operazione del gestore code, il gestore code scrive un messaggio nel log degli errori. Rieseguire l'operazione non riuscita ogni 10 minuti fino a quando l'operazione non ha esito positivo. Per impostazione predefinita, si dispone di cinque giorni per gestire l'errore; in caso contrario, il gestore code scrive un messaggio nel log degli errori e si arresta. Puoi posticipare la chiusura di cinque giorni.
2. Se l'errore compromette l'operazione del gestore code, quest' ultimo scrive un messaggio nel log degli errori e si arresta immediatamente.

Un errore che compromette il funzionamento del gestore code è un errore che il gestore code non è stato in grado di diagnosticare o un errore che potrebbe avere conseguenze imprevedibili. Questo tipo di errore spesso si verifica quando il gestore code scrive un file FFST . Gli errori che compromettono il funzionamento del gestore code potrebbero essere causati da un bug in IBM MQ, o da un amministratore o da un programma, che esegue operazioni impreviste, come terminare un processo IBM MQ .

Il punto della modifica nel comportamento di ripristino da errore è quello di limitare il tempo in cui il gestore code continua l'esecuzione con un numero crescente di definizioni cluster incongruenti. Con l'aumentare del numero di incongruenze nelle definizioni di cluster, aumenta la probabilità di un comportamento anomalo dell'applicazione.

La scelta predefinita di arrestare il gestore code dopo cinque giorni rappresenta un compromesso tra la limitazione del numero di incongruenze e la disponibilità del gestore code fino a quando non vengono rilevati e risolti i problemi.

È possibile estendere il tempo prima che il gestore code venga arrestato per un periodo di tempo indefinito, mentre si corregge il problema o si attende la chiusura pianificata del gestore code. La permanenza di cinque giorni mantiene il gestore code in esecuzione per un lungo fine settimana, dandoti il tempo per reagire a eventuali problemi o prolungare il tempo prima di riavviare il gestore code.

Azioni correttive

Si dispone di una scelta di azioni per gestire i problemi di ripristino degli errori cluster. La prima scelta è quella di monitorare e correggere il problema e la seconda di monitorare e posticipare la correzione del problema.

1. Monitorare il log degli errori del gestore code per i messaggi di errore [AMQ9448](#) e [AMQ5008](#) e correggere il problema.

[AMQ9448](#) indica che il gestore repository ha restituito un errore dopo l'esecuzione di un comando. Questo errore contrassegna l'inizio di un nuovo tentativo di esecuzione del comando ogni 10 minuti ed eventualmente l'arresto del gestore code dopo cinque giorni, a meno che l'arresto non venga posticipato.

[AMQ5008](#) indica che il gestore code è stato arrestato perché manca un processo IBM MQ. [AMQ5008](#) risulta dall'arresto del gestore repository dopo cinque giorni. Se il gestore repository si arresta, il gestore code si arresta.

2. Monitorare il log degli errori del gestore code per il messaggio di errore [AMQ9448](#) e posticipare la correzione del problema.

Se si disabilita il richiamo dei messaggi da `SYSTEM.CLUSTER.COMMAND.QUEUE`, il gestore repository interrompe il tentativo di eseguire i comandi e continua indefinitamente senza elaborare alcun lavoro. Tuttavia, tutti gli handle che il gestore repository detiene nelle code vengono rilasciati. Poiché il gestore repository non viene arrestato, il gestore code non viene arrestato dopo cinque giorni.

Eseguire un comando MQSC per disabilitare il richiamo dei messaggi da `SYSTEM.CLUSTER.COMMAND.QUEUE`:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
```

Per riprendere la ricezione dei messaggi da `SYSTEM.CLUSTER.COMMAND.QUEUE` eseguire un comando MQSC:

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

Considerazione speciale

L'arresto di `amqrrmf` in IBM MQ provoca l'arresto del gestore code, poiché viene considerato un malfunzionamento del gestore code. È necessario arrestare il processo `amqrrmf` solo se si imposta il parametro di ottimizzazione del gestore code, `TolerateRepositoryFailure`.

Esempio

```
TuningParameters:  
  TolerateRepositoryFailure=TRUE
```

Figura 86. Impostare `TolerateRepositoryFailure` su `TRUE` in `qm.ini`

Concetti correlati

[“File di configurazione del gestore code, `qm.ini`” a pagina 107](#)

Un file di configurazione del gestore code, `qm.ini`, contiene informazioni relative a uno specifico gestore code. Gli attributi che è possibile utilizzare per modificare la configurazione di un singolo gestore code sovrascrivono le impostazioni per IBM MQ.

Configurazione di risorse di JMS e Jakarta Messaging

Uno dei modi in cui un'applicazione JMS o Jakarta Messaging può creare e configurare le risorse necessarie per connettersi a IBM MQ e accedere alle destinazioni per l'invio o la ricezione di messaggi è utilizzando JNDI (Java Naming and Directory Interface) per richiamare gli oggetti gestiti da un'ubicazione all'interno del servizio di denominazione e di directory denominato spazio nomi JNDI. Prima che un'applicazione JMS possa richiamare gli oggetti gestiti da uno spazio nomi JNDI, è necessario creare e configurare gli oggetti gestiti.

Informazioni su questa attività

JM 3.0 Da IBM MQ 9.3.0, Jakarta Messaging 3.0 è supportato per lo sviluppo di nuove applicazioni. IBM MQ 9.3.0 e successive continuano a supportare JMS 2.0 per le applicazioni esistenti. Non è supportato utilizzare sia l'API Jakarta Messaging 3.0 che l'API JMS 2.0 nella stessa applicazione. Per ulteriori informazioni, fare riferimento a [Utilizzo delle classi IBM MQ per JMS/Jakarta Messaging](#).

È possibile creare e configurare gli oggetti gestiti in IBM MQ utilizzando uno dei seguenti strumenti:

Strumenti di gestione di IBM MQ JMS e Jakarta Messaging

Lo strumento di amministrazione IBM MQ JMS, **JMSAdmin**, e lo strumento di amministrazione Jakarta Messaging, **JMS30Admin**, sono strumenti della riga comandi che è possibile utilizzare per creare e configurare gli oggetti IBM MQ JMS e Jakarta Messaging memorizzati in LDAP, in un file system locale o in altre ubicazioni. Gli strumenti di gestione JMS e Jakarta Messaging utilizzano una sintassi simile a **runmqsc** e supportano anche gli script.

Gli strumenti di amministrazione utilizzano un file di configurazione per impostare i valori di determinate proprietà. Viene fornito un file di configurazione di esempio, che è possibile modificare per adattarlo al sistema prima di iniziare utilizzando lo strumento per configurare le risorse JMS. Per ulteriori informazioni sul file di configurazione, consultare [“Configurazione degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 713.

JMS 2.0 IBM MQ Explorer

Per JMS 2.0, è possibile utilizzare IBM MQ Explorer per creare e gestire le definizioni di oggetti JMS 2.0 memorizzate in LDAP, in un file system locale o in altre ubicazioni.

JM 3.0 Per Jakarta Messaging 3.0, non è possibile gestire JNDI utilizzando IBM MQ Explorer. La gestione JNDI è supportata dalla variante Jakarta Messaging 3.0 di **JMSAdmin**, che è **JMS30Admin**.

Le applicazioni IBM MQ JMS distribuite in WebSphere Application Server devono accedere agli oggetti JMS dal repository JNDI del server delle applicazioni. Pertanto, se si utilizza la messaggistica JMS tra WebSphere Application Server e IBM MQ, è necessario creare oggetti in WebSphere Application Server che corrispondono agli oggetti creati in IBM MQ.

JM 3.0 Sebbene IBM MQ 9.3 e versioni successive supportino Jakarta Messaging 3.0, WebSphere Application Server non dispone attualmente di un supporto equivalente. Pertanto, in WebSphere Application Server, si configurano risorse Java Message Service 2.0.

IBM MQ Explorer e lo strumento di amministrazione IBM MQ JMS non possono essere utilizzati per gestire oggetti IBM MQ JMS memorizzati in WebSphere Application Server. Invece, è possibile creare e configurare gli oggetti gestiti in WebSphere Application Server utilizzando uno dei seguenti strumenti:

WebSphere Application Server Console di gestione

La console di amministrazione WebSphere Application Server è uno strumento basato sul web che è possibile utilizzare per gestire gli oggetti di IBM MQ JMS in WebSphere Application Server.

Client di script wsadmin WebSphere Application Server

Il client di script WebSphere Application Server wsadmin fornisce comandi specializzati per la gestione di oggetti IBM MQ JMS in WebSphere Application Server.

Se si desidera utilizzare un'applicazione JMS per accedere alle risorse di un gestore code IBM MQ da WebSphere Application Server, è necessario utilizzare il provider di messaggistica IBM MQ in WebSphere Application Server, che contiene una versione di IBM MQ classes for JMS. L'adattatore di risorse IBM MQ fornito con WebSphere Application Server viene utilizzato da tutte le applicazioni che eseguono la messaggistica JMS con il fornitore di messaggistica IBM MQ. L'adattatore di risorse IBM MQ viene di solito aggiornato automaticamente quando si applicano i fix pack WebSphere Application Server, ma se l'adattatore di risorse è stato precedentemente aggiornato manualmente, è necessario aggiornare manualmente la propria configurazione per garantire che la manutenzione venga applicata correttamente.

Concetti correlati

[Creazione e configurazione di factory di connessione e destinazioni in un'applicazione IBM MQ per JMS](#)

Riferimenti correlati

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

Configurazione di factory di connessione e destinazioni in un namespace JNDI

Le applicazioni JMS e Jakarta Messaging accedono agli oggetti gestiti nel servizio di denominazione e directory tramite JNDI (Java Naming and Directory Interface). Gli oggetti gestiti JMS o Jakarta Messaging vengono memorizzati in un'ubicazione all'interno del servizio di denominazione e di directory a cui si fa riferimento come spazio dei nomi JNDI. Un'applicazione JMS o Jakarta Messaging può ricercare gli oggetti gestiti per connettersi a IBM MQ e accedere alle destinazioni per inviare o ricevere messaggi.

Informazioni su questa attività

Le applicazioni JMS o Jakarta Messaging cercano i nomi degli oggetti JMS o Jakarta Messaging nel servizio di denominazione e di directory utilizzando i contesti:

Contesto iniziale

Il contesto iniziale definisce la radice dello spazio nomi JNDI. Per ogni ubicazione nel servizio di denominazione e directory, è necessario specificare un contesto iniziale per fornire un punto di partenza da cui un'applicazione JMS o Jakarta Messaging può risolvere i nomi degli oggetti gestiti in tale ubicazione del servizio di denominazione e di directory.

Contesti secondari

Un contesto può avere uno o più contesti secondari. Un contesto secondario è una suddivisione di uno spazio nomi JNDI e può contenere oggetti gestiti come factory di connessione e destinazioni, nonché altri contesti secondari. Un contesto secondario non è proprio un oggetto; è semplicemente un'estensione della convenzione di denominazione per gli oggetti nel contesto secondario.

Prima che un'applicazione IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging possa richiamare gli oggetti gestiti da uno spazio dei nomi JNDI, è necessario prima creare gli oggetti gestiti. È possibile creare e configurare i seguenti tipi di oggetto JMS o Jakarta Messaging :

Factory connessioni

Un oggetto factory di connessione JMS o Jakarta Messaging definisce una serie di proprietà di configurazione standard per le connessioni. Un'applicazione JMS o Jakarta Messaging utilizza una factory di connessione per creare una connessione a IBM MQ. È possibile creare una factory di connessione specifica per uno dei due domini di messaggistica, il dominio di messaggistica point-to-point e il dominio di messaggistica di pubblicazione / sottoscrizione.

In alternativa, da JMS 1.1, è possibile creare factory di connessione indipendenti dal dominio che possono essere utilizzati sia per la messaggistica point-to-point che per la messaggistica di pubblicazione / sottoscrizione. Per ulteriori informazioni, consultare [Il modello JMS e Jakarta Messaging](#).

Destinazione

Una destinazione JMS o Jakarta Messaging è un oggetto che rappresenta la destinazione dei messaggi prodotti dal client e l'origine dei messaggi utilizzati da un'applicazione JMS . L'applicazione JMS o Jakarta Messaging può utilizzare un singolo oggetto di destinazione per inserire e ricevere messaggi oppure l'applicazione può utilizzare oggetti di destinazione separati. Esistono due tipi di oggetto di destinazione:

- Destinazione coda JMS o Jakarta Messaging utilizzata nella messaggistica point - to - point
- Destinazione argomento JMS o Jakarta Messaging utilizzata nella messaggistica di pubblicazione / sottoscrizione

JMS 2.0 Per JMS 2.0, è possibile creare contesti e oggetti gestiti utilizzando IBM MQ Explorer o lo strumento di amministrazione IBM MQ JMS **JMSAdmin**.

Nota: **JM 3.0** Per Jakarta Messaging 3.0, non è possibile gestire JNDI utilizzando IBM MQ Explorer. La gestione JNDI è supportata dalla variante Jakarta Messaging 3.0 di **JMSAdmin**, che è **JMS30Admin**.

Il seguente diagramma mostra un esempio di oggetti JMS o Jakarta Messaging creati in uno spazio dei nomi JNDI IBM MQ .

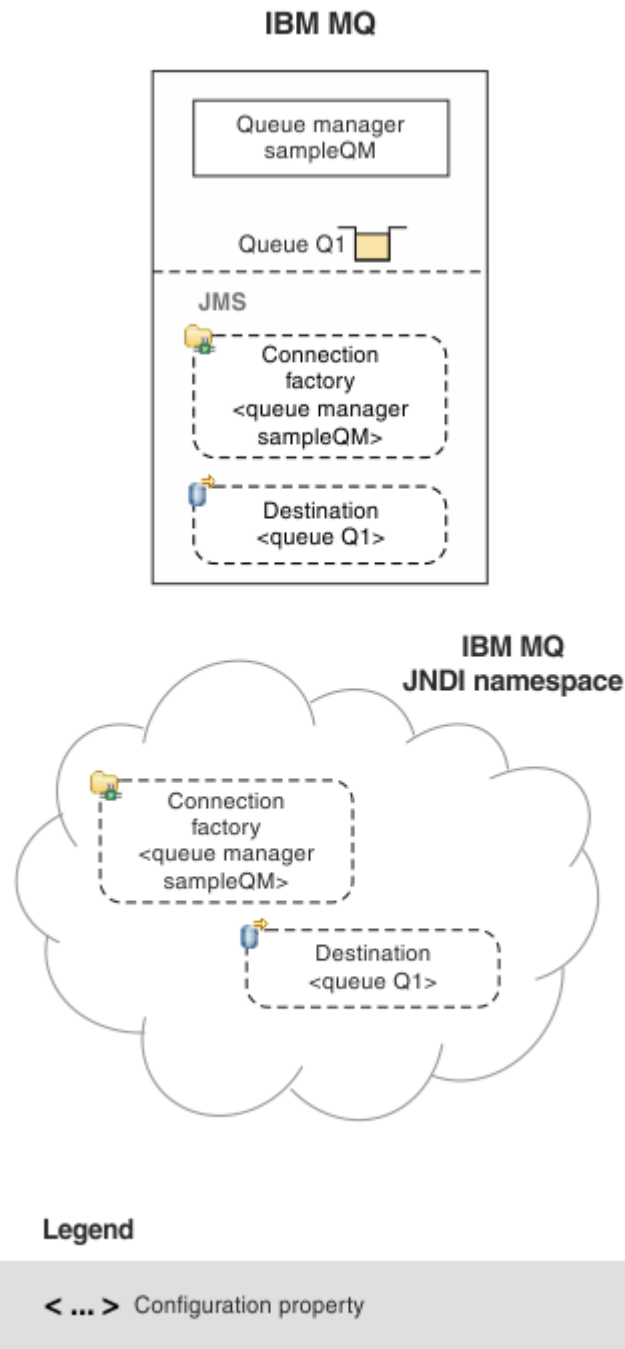


Figura 87. Oggetti JMS o Jakarta Messaging creati in IBM MQ

Se si utilizza la messaggistica JMS tra WebSphere Application Server e IBM MQ, è necessario creare gli oggetti corrispondenti in WebSphere Application Server da utilizzare per comunicare con IBM MQ. Quando si crea uno di questi oggetti in WebSphere Application Server, viene memorizzato nello spazio dei nomi JNDI WebSphere Application Server come mostrato nel seguente diagramma.

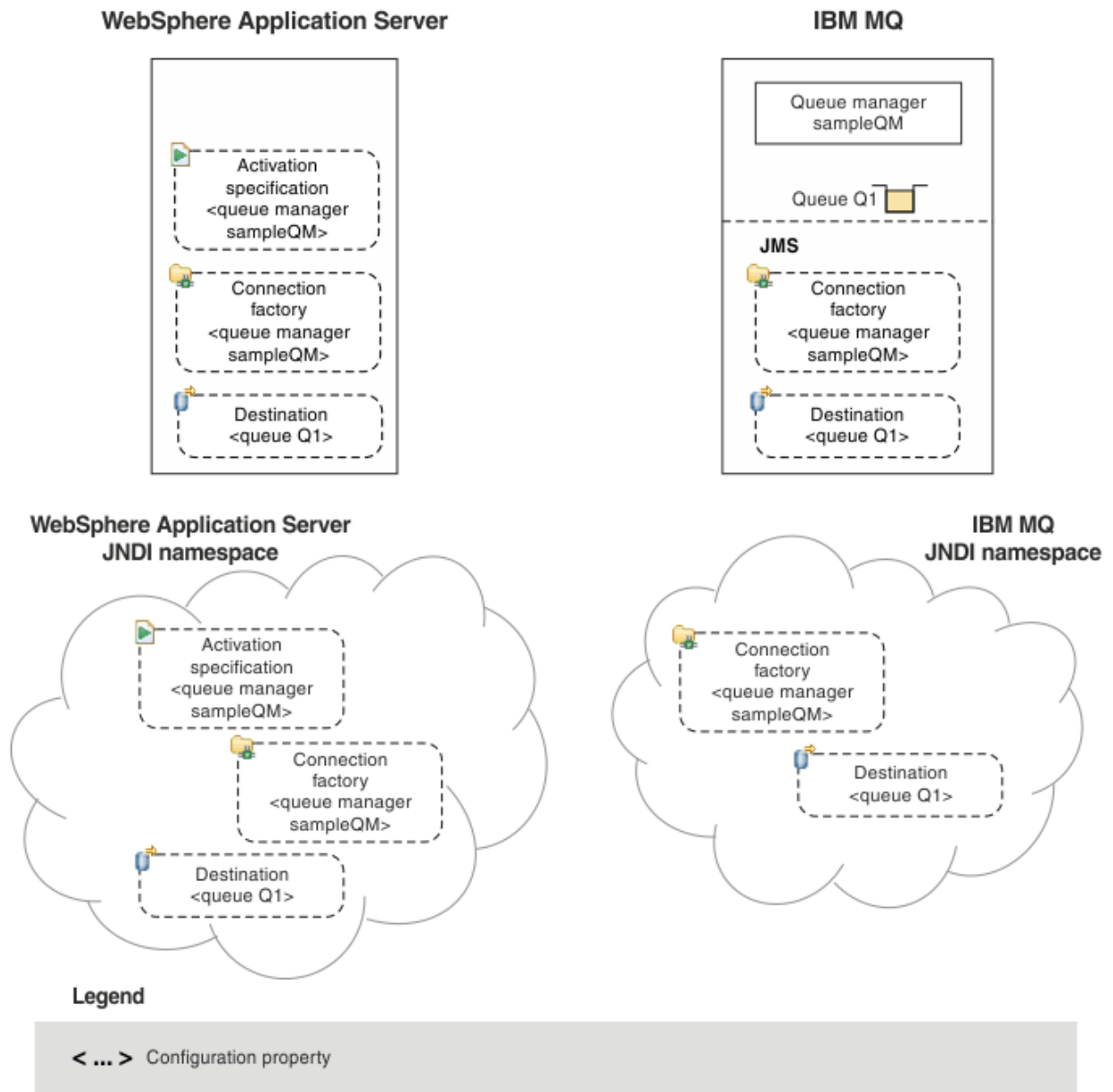


Figura 88. Oggetti creati in WebSphere Application Server e oggetti corrispondenti in IBM MQ

JM 3.0 Sebbene IBM MQ 9.3 e versioni successive supportino Jakarta Messaging 3.0, WebSphere Application Server non dispone attualmente di un supporto equivalente. Pertanto, in WebSphere Application Server, si configurano risorse Java Message Service 2.0.

Se l'applicazione utilizza un MDB (message - driven bean), il factory di connessione viene utilizzato solo per i messaggi in uscita e i messaggi in entrata vengono ricevuti da una specifica di attivazione. Le specifiche di attivazione fanno parte dello standard Java EE Connector Architecture 1.5 (JCA 1.5). JCA 1.5 fornisce un modo standard per integrare i provider JMS, come IBM MQ, con i server delle applicazioni Java EE come WebSphere Application Server. Una specifica di attivazione JMS può essere associata a uno o più MDB (message driven bean) e fornisce la configurazione necessaria affinché questi MDB siano in ascolto dei messaggi che arrivano a una destinazione.

È possibile utilizzare la console di gestione WebSphere Application Server o i comandi di script wsadmin per creare e configurare le risorse JMS necessarie.

Procedura

- **JMS 2.0**
Per configurare gli oggetti JMS per IBM MQ utilizzando IBM MQ Explorer, consultare [“Configurazione di oggetti JMS 2.0 utilizzando IBM MQ Explorer”](#) a pagina 710.
- **JMS 2.0**
Per configurare gli oggetti JMS per IBM MQ utilizzando lo strumento di gestione IBM MQ JMS , **JMSAdmin**, consultare [“Configurazione degli oggetti JMS e Jakarta Messaging utilizzando gli strumenti di amministrazione”](#) a pagina 711.
- **JM 3.0**
Per configurare gli oggetti Jakarta Messaging per IBM MQ utilizzando lo strumento di gestione IBM MQ Jakarta Messaging , **JMS30Admin**, consultare [“Configurazione degli oggetti JMS e Jakarta Messaging utilizzando gli strumenti di amministrazione”](#) a pagina 711.
- **JMS 2.0**
Per configurare gli oggetti JMS per WebSphere Application Server, consultare [“Configurazione di risorse JMS 2.0 in WebSphere Application Server”](#) a pagina 721.

Risultati

Un'applicazione IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging può richiamare gli oggetti gestiti dallo spazio dei nomi JNDI e, se necessario, impostare o modificare una o più delle relative proprietà utilizzando le estensioni IBM JMS o IBM MQ JMS .

Attività correlate

Utilizzo di JNDI per richiamare gli oggetti gestiti in una applicazione JMS

Creazione e configurazione di factory di connessione e destinazioni in una applicazione IBM MQ classes for JMS

JMS 2.0 Configurazione di oggetti JMS 2.0 utilizzando IBM MQ Explorer

Utilizzare la GUI (graphical user interface) IBM MQ Explorer per creare oggetti JMS da oggetti IBM MQ e oggetti IBM MQ da oggetti JMS , nonché per la gestione e il monitoraggio di altri oggetti IBM MQ .

Informazioni su questa attività

JMS 2.0 IBM MQ Explorer è la GUI (graphical user interface) in cui è possibile amministrare e controllare gli oggetti IBM MQ, siano essi ospitati dal computer locale o su un sistema remoto. IBM MQ Explorer viene eseguito su Windows e Linux for x86-64. È possibile collegarlo in remoto ai gestori code in esecuzione su qualsiasi piattaforma supportata, incluso z/OS, consentendo la visualizzazione, esplorazione e modifica del backbone di messaggistica dalla console.

Nota: **JM 3.0** Per Jakarta Messaging 3.0, non è possibile gestire JNDI utilizzando IBM MQ Explorer. La gestione JNDI è supportata dalla variante Jakarta Messaging 3.0 di **JMSAdmin**, che è **JMS30Admin**.

In IBM MQ Explorer, tutte le factory di connessione vengono memorizzate nelle cartelle Factory di connessione nel contesto appropriato e nei contesti secondari.

È possibile eseguire i seguenti tipi di attività con IBM MQ Explorer, contestualmente da un oggetto esistente in IBM MQ Explorero dall'interno di una procedura guidata di creazione di un nuovo oggetto:

- Creare un factory di connessione JMS da uno dei seguenti oggetti IBM MQ :
 - Un gestore code IBM MQ , sul computer locale o su un sistema remoto.
 - Un canale IBM MQ .
 - Un listener IBM MQ .
- Aggiungere un gestore code IBM MQ a IBM MQ Explorer utilizzando un factory di connessione JMS .

- Creare una coda JMS da una coda IBM MQ .
- Creare una coda IBM MQ da una coda JMS .
- Creare un argomento JMS da un argomento IBM MQ , che può essere un oggetto IBM MQ o un argomento dinamico.
- Creare un argomento IBM MQ da un argomento JMS .

Procedura

- Avviare IBM MQ Explorer, se non è già in esecuzione.
Se IBM MQ Explorer è in esecuzione e visualizza la pagina di benvenuto, chiudere la pagina di benvenuto per avviare la gestione degli oggetti IBM MQ .
- Se non è già stato fatto, creare un contesto iniziale definendo la root dello spazio dei nomi JNDI in cui gli oggetti JMS sono memorizzati nel servizio di denominazione e di directory.
Dopo aver aggiunto il contesto iniziale a IBM MQ Explorer, è possibile creare oggetti factory di connessione, oggetti destinazione e contesti secondari nello spazio nomi JNDI.
Il contesto iniziale viene visualizzato nella vista Navigator nella cartella Oggetti amministrati JMS .
Tenere presente che, sebbene venga visualizzato il contenuto completo dello spazio dei nomi JNDI, in IBM MQ Explorer è possibile modificare solo gli oggetti IBM MQ classes for JMS memorizzati in tale spazio. Per ulteriori informazioni, fare riferimento a [Aggiunta di un contesto iniziale](#).
- Creare e configurare i contesti secondari e gli oggetti amministrati JMS necessari.
Per ulteriori informazioni, consultare [Creazione e configurazione di oggetti amministrati JMS](#).
- Configurare IBM MQ.
Per ulteriori informazioni, consultare [Configurazione di IBM MQ utilizzando IBM MQ Explorer](#) .

Concetti correlati

[Introduzione a IBM MQ Explorer](#)

[Creazione e configurazione di factory di connessione e destinazioni in una applicazione IBM MQ classes for JMS](#)

Configurazione degli oggetti JMS e Jakarta Messaging utilizzando gli strumenti di amministrazione

IBM MQ fornisce strumenti di amministrazione che è possibile utilizzare per definire le proprietà di otto tipi di oggetti IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging e per memorizzarli all'interno di uno spazio dei nomi JNDI. Le applicazioni possono quindi utilizzare JNDI per richiamare questi oggetti gestiti dallo spazio dei nomi.

Informazioni su questa attività

JMS 2.0 per [JMS 2.0](#), l'amministrazione JNDI è supportata dallo strumento **JMSAdmin** .

JM 3.0 Per [Jakarta Messaging 3.0](#), la gestione JNDI è supportata dalla variante Jakarta Messaging 3.0 di **JMSAdmin**, che è **JMS30Admin**.

La seguente tabella mostra gli otto tipi di oggetti gestiti che è possibile creare, configurare e manipolare utilizzando i verbi. La colonna Parola chiave mostra le stringhe che è possibile sostituire per *TYPE* nei comandi mostrati in [Tabella 37 a pagina 712](#).

Tabella 37. I tipi di oggetto JMS e Jakarta Messaging gestiti dallo strumento di amministrazione

Tipo oggetto	Parola chiave	Descrizione
MQConnectionFactory	CF	L'implementazione IBM MQ dell'interfaccia JMS ConnectionFactory . Questo rappresenta un oggetto factory per la creazione di connessioni in entrambi i domini point-to-point e di pubblicazione / sottoscrizione.
Factory di MQQueueConnection	QCF	L'implementazione IBM MQ dell'interfaccia factory JMS QueueConnection. Rappresenta un oggetto factory per la creazione di connessioni nel dominio point - to - point.
Factory MQTopicConnection	TCF	L'implementazione IBM MQ dell'interfaccia factory di JMS TopicConnection. Rappresenta un oggetto factory per la creazione di connessioni nel dominio di pubblicazione / sottoscrizione.
MQQUEUE	Q	L'implementazione IBM MQ dell'interfaccia della coda JMS . Rappresenta una destinazione per i messaggi nel dominio point - to - point.
MQArgomento	T	L'implementazione IBM MQ dell'interfaccia dell'argomento JMS . Rappresenta una destinazione per i messaggi nel dominio di pubblicazione / sottoscrizione.
MQXAConnectionFactory “1” a pagina 712	XACF	L'implementazione IBM MQ dell'interfaccia JMS XAConnectionFactory . Rappresenta un oggetto factory per la creazione di connessioni nei domini point-to-point e di pubblicazione / sottoscrizione e dove le connessioni utilizzano le versioni XA delle classi JMS .
MQXAQueueConnectionFactory “1” a pagina 712	XAQCF	L'implementazione IBM MQ dell'interfaccia factory JMS XAQueueConnection. Ciò rappresenta un oggetto factory per creare connessioni nel dominio point - to - point che utilizzano le versioni XA delle classi JMS .
MQXATopicConnectionFactory “1” a pagina 712	XATCF	L'implementazione IBM MQ dell'interfaccia factory JMS XATopicConnection. Rappresenta un oggetto factory per la creazione di connessioni nel dominio di pubblicazione / sottoscrizione che utilizzano le versioni XA delle classi JMS .

Nota:

1. Queste classi vengono fornite per essere utilizzate dai fornitori dei server delle applicazioni. È improbabile che siano direttamente utili per i programmatori di applicazioni.

Per ulteriori informazioni su come configurare questi oggetti, consultare [“Configurazione di oggetti JMS” a pagina 721.](#)

I valori e i tipi di proprietà necessari per utilizzare questo strumento sono elencati in [Proprietà degli oggetti IBM MQ classes for JMS](#).

È anche possibile utilizzare lo strumento per manipolare i contesti secondari dello spazio nomi della directory all'interno di JNDI come descritto in [“Configurazione dei contesti secondari” a pagina 717](#).

JMS 2.0 Per JMS 2.0 e versioni precedenti, è anche possibile creare e configurare gli oggetti gestiti IBM MQ classes for JMS con IBM MQ Explorer.

JM 3.0 Per Jakarta Messaging 3.0, non è possibile gestire JNDI utilizzando IBM MQ Explorer. La gestione JNDI è supportata dalla variante Jakarta Messaging 3.0 di **JMSAdmin**, che è **JMS30Admin**.

Concetti correlati

[Creazione e configurazione di factory di connessione e destinazioni in una applicazione IBM MQ classes for JMS](#)

[Utilizzo di JNDI per richiamare gli oggetti gestiti in una applicazione JMS](#)

Configurazione degli strumenti JMSAdmin e JMS30Admin

Gli strumenti di amministrazione IBM MQ JMS e Jakarta Messaging utilizzano un file di configurazione per impostare i valori di alcune proprietà. In ogni caso, viene fornito un file di configurazione di esempio che è possibile modificare per adattarlo al sistema.

Informazioni su questa attività

JM 3.0 IBM MQ 9.3.0 ha introdotto il supporto per [Jakarta Messaging 3.0](#). JMS 2.0 è ancora completamente supportato.

Il file di configurazione è un file di testo semplice costituito da una serie di coppie chiave - valore, separate dal segno uguale (=). Configurare lo strumento di amministrazione impostando i valori per le tre proprietà definite nel file di configurazione. Il seguente esempio mostra queste tre proprietà:

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

In questo esempio, un segno cancelletto (#) nella prima colonna della riga indica un commento o una riga non utilizzata.

Un file di configurazione di esempio, utilizzato come file di configurazione predefinito, viene fornito con IBM MQ. Il file di esempio è denominato `JMSAdmin.config` (per JMS 2.0) o `JMS30Admin.config` (per Jakarta Messaging 3.0). Questo file si trova nella directory `MQ_JAVA_INSTALL_PATH/bin`. È possibile modificare il file di esempio per definire le impostazioni necessarie per il proprio sistema oppure creare il proprio file di configurazione.

Quando si avvia lo strumento di gestione, è possibile specificare il file di configurazione da utilizzare utilizzando il parametro della riga comandi `-cfg`, come descritto in [“Avvio degli strumenti JMSAdmin e JMS30Admin” a pagina 715](#). Se non si specifica un nome file di configurazione quando si richiama lo strumento, lo strumento tenta di caricare il file di configurazione predefinito (`JMSAdmin.config` o `JMS30Admin.config`). Ricerca questo file prima nella directory corrente e poi nella directory `MQ_JAVA_INSTALL_PATH/bin`, dove `MQ_JAVA_INSTALL_PATH` è il percorso per l'installazione di IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging.

I nomi degli oggetti JMS o Jakarta Messaging memorizzati in un ambiente LDAP devono essere conformi alle convenzioni di denominazione LDAP. Una di queste convenzioni è che i nomi oggetto e contesto devono includere un prefisso, come `cn=` (nome comune) o `ou=` (unità organizzativa). Lo strumento di amministrazione semplifica l'utilizzo dei provider di servizi LDAP consentendo di fare riferimento a nomi di oggetti e di contesto senza un prefisso. Se non si fornisce un prefisso, lo strumento aggiunge

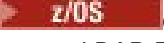
automaticamente un prefisso predefinito al nome fornito. Per LDAP, è cn=. Se necessario, è possibile modificare il prefisso predefinito impostando la proprietà **NAME_PREFIX** nel file di configurazione.

Nota: Potrebbe essere necessario configurare il server LDAP per memorizzare gli oggetti Java . Per ulteriori informazioni, consultare la documentazione per il server LDAP.

Procedura

1. Definire il provider di servizi utilizzato dallo strumento configurando la proprietà **INITIAL_CONTEXT_FACTORY** .

I valori supportati per questa proprietà sono i seguenti:

- com.sun.jndi.ldap.LdapCtxFactory (per LDAP)
- com.sun.jndi.fscontext.RefFSContextFactory (per il contesto del file system)
-  com.ibm.jndi.LDAPCtxFactory è supportato solo su z/OS e fornisce l'accesso a un server LDAP. Tuttavia, questa classe non è compatibile con com.sun.jndi.ldap.LdapCtxFactory, poiché gli oggetti creati utilizzando un factory InitialContext non possono essere letti o modificati utilizzando l'altro factory.

È anche possibile utilizzare lo strumento di amministrazione per connettersi ad altri contesti JNDI utilizzando tre parametri definiti nel file di configurazione JMSAdmin o JMS30Admin . Per utilizzare un factory InitialContext differente:

- a) Impostare la proprietà **INITIAL_CONTEXT_FACTORY** sul nome classe richiesto.
- b) Definire il funzionamento del factory InitialContext utilizzando le proprietà **USE_INITIAL_DIR_CONTEXT**, **NAME_PREFIX** e **NAME_READABILITY_MARKER** .

Le impostazioni per queste proprietà sono descritte nei commenti del file di configurazione di esempio.

Non è necessario definire le proprietà **USE_INITIAL_DIR_CONTEXT**, **NAME_PREFIX** e **NAME_READABILITY_MARKER** se si utilizza uno dei valori **INITIAL_CONTEXT_FACTORY** supportati. Tuttavia, è possibile assegnare dei valori a queste proprietà se si desidera sovrascrivere i valori predefiniti del sistema. Ad esempio, se gli oggetti sono archiviati in un ambiente LDAP, è possibile modificare il prefisso predefinito che lo strumento aggiunge ai nomi oggetto e contesto impostando la proprietà **NAME_PREFIX** sul prefisso richiesto.

Se si omettono una o più delle tre proprietà InitialContextFactory, lo strumento di amministrazione fornisce valori predefiniti appropriati in base ai valori delle altre proprietà.

2. Definire l'URL del contesto iniziale della sessione configurando la proprietà **PROVIDER_URL** .

Questo URL è la root di tutte le operazioni JNDI eseguite dallo strumento. Sono supportate due forme di questa proprietà:

- ldap://hostname/contextname
- file: [unità:] /nomepercorso

Il formato dell'URL LDAP può variare, a seconda del provider LDAP. Per ulteriori informazioni, consultare la documentazione LDAP.

3. Definire se JNDI trasmette le credenziali di sicurezza al provider di servizi configurando la proprietà **SECURITY_AUTHENTICATION** .

Questa proprietà viene utilizzata solo quando viene utilizzato un fornitore di servizi LDAP e può assumere uno dei tre seguenti valori:

none (autenticazione anonima)

Se si imposta questo parametro su none, JNDI non passa alcuna credenziale di sicurezza al provider di servizi e viene eseguita l' *autenticazione anonima* .

semplice (autenticazione semplice)

Se si imposta il parametro su semplice, le credenziali di sicurezza vengono trasmesse tramite JNDI al fornitore del servizio sottostante. Queste credenziali di sicurezza sono nel formato di un DN (distinguished name) utente e parola d'ordine.

CRAM-MD5 (meccanismo di autenticazione CRAM-MD5)

Se si imposta il parametro su CRAM-MD5, le credenziali di sicurezza vengono trasmesse tramite JNDI al provider di servizi sottostante. Queste credenziali di sicurezza sono nel formato di un DN (distinguished name) utente e parola d'ordine.

Se non si fornisce un valore valido per la proprietà **SECURITY_AUTHENTICATION**, la proprietà assume il valore predefinito none.

Se le credenziali di sicurezza sono richieste, vengono richieste quando lo strumento viene inizializzato. È possibile evitare questa situazione impostando le proprietà **PROVIDER_USERDN** e **PROVIDER_PASSWORD** nel file di configurazione JMSAdmin.

Nota: Se non si utilizzano queste proprietà, il testo immesso, *inclusa la password*, viene ripetuto sullo schermo. Ciò potrebbe avere implicazioni sulla sicurezza.

Lo strumento non esegue l'autenticazione; l'attività di autenticazione è delegata al server LDAP. L'amministratore del server LDAP deve impostare e gestire i privilegi di accesso a parti differenti della directory. Per ulteriori informazioni, consultare la documentazione LDAP. Se l'autenticazione ha esito negativo, lo strumento visualizza un messaggio di errore appropriato e termina.

Informazioni più dettagliate sulla sicurezza e JNDI sono disponibili nella documentazione sul sito Web Oracle Java ([Oracle Technology Network for Java Developers](#)).

Avvio degli strumenti JMSAdmin e JMS30Admin

Gli strumenti di amministrazione di Jakarta Messaging e IBM MQ JMS dispongono di una CLI (command-line interface) che è possibile utilizzare in modo interattivo o per avviare un processo batch.

Informazioni su questa attività

La modalità interattiva fornisce un prompt dei comandi in cui è possibile immettere i comandi di gestione. In modalità batch, il comando per avviare lo strumento include il nome di un file contenente uno script di comandi di amministrazione.

Procedura

Modalità interattiva

- Per avviare lo strumento in modalità interattiva, immettere il seguente comando:

```
> JMS 2.0
```

```
JMSAdmin [-t] [-v] [-cfg config_filename]
```

```
> JM 3.0
```

```
JMS30Admin [-t] [-v] [-cfg config_filename]
```

dove:

-t

Abilita la traccia (il valore predefinito è traccia disattivata).

Il file di traccia viene creato in "%MQ_JAVA_DATA_PATH%\errors (Windows) o /var/mqm/trace (AIX and Linux). Il nome del file di traccia è nel formato:

```
mqjms_PID.trc
```

dove *PID* è l'ID processo della JVM.

-v

Produce un output dettagliato (il valore predefinito è un output conciso).

-cfg nomefile_config

Denomina un file di configurazione alternativo. Se questo parametro viene omissso, viene utilizzato il file di configurazione predefinito, `JMSAdmin.config` (per JMS 2.0) o `JMS30Admin.config` (per Jakarta Messaging 3.0). Per ulteriori informazioni sul file di configurazione, consultare [“Configurazione degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 713.

Viene visualizzato un prompt dei comandi, che indica che lo strumento è pronto ad accettare i comandi di gestione. Questa richiesta inizialmente viene visualizzata come:

```
InitCtx>
```

che indica che il contesto corrente (ossia, il contesto JNDI a cui fanno riferimento tutte le operazioni di denominazione e di directory) è il contesto iniziale definito nel parametro di configurazione **PROVIDER_URL**. Per maggiori informazioni su questo parametro, fare riferimento a [“Configurazione degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 713.

Mentre si attraversa lo spazio dei nomi della directory, il prompt cambia per riflettere questo, in modo che il prompt visualizzi sempre il contesto corrente.

Modalità batch

- Per avviare lo strumento in modalit ... batch, immettere il seguente comando:

```
JMS 2.0
```

```
JMSAdmin test.scp
```

```
JM 3.0
```

```
JMS30Admin test.scp
```

dove `test.scp` è un file script che contiene i comandi di gestione. Per ulteriori informazioni, consultare [“Utilizzo dei comandi di gestione con JMSAdmin e JMS30Admin”](#) a pagina 716. L'ultimo comando nel file deve essere il comando END.

Utilizzo dei comandi di gestione con JMSAdmin e JMS30Admin

Gli strumenti di amministrazione IBM MQ JMS e Jakarta Messaging accettano comandi costituiti da un verbo di amministrazione e i relativi parametri appropriati.

Informazioni su questa attività

La seguente tabella elenca i verbi di amministrazione che è possibile utilizzare quando si immettono comandi con gli strumenti di gestione.

Verbo	Forma breve	Descrizione
MODIFICA	Alt	Modificare almeno una delle proprietà di un oggetto gestito
Definisci	DEF	Creare e memorizzare un oggetto gestito o creare un contesto secondario
VISUALIZZA	DIS	Visualizza le proprietà di uno o più oggetti gestiti memorizzati o il contenuto del contesto corrente
ELIMINA	DEL	Rimuovere uno o più oggetti gestiti dallo spazio dei nomi o rimuovere un contesto secondario vuoto

Tabella 38. Verbi di amministrazione (Continua)

Verbo	Forma breve	Descrizione
MODIFICA	CHG	Modificare il contesto corrente, consentendo all'utente di attraversare lo spazio dei nomi della directory in un punto qualsiasi al di sotto del contesto iniziale (autorizzazione di sicurezza in sospenso)
Copia	C.A.P.	Creare una copia di un oggetto gestito memorizzato, memorizzandolo con un nome alternativo
SPOSTA	MV	Modificare il nome con cui è memorizzato un oggetto gestito
FINE		Chiudere lo Strumento di amministrazione

Procedura

- Se lo strumento di amministrazione non è già stato avviato, avviarlo come descritto in [“Avvio degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 715.

Viene visualizzato il prompt dei comandi, che indica che lo strumento è pronto ad accettare i comandi di gestione. Questa richiesta inizialmente viene visualizzata come:

```
InitCtx>
```

Per modificare il contesto corrente, utilizzare il comando CHANGE come descritto in [“Configurazione dei contesti secondari”](#) a pagina 717.

- Immettere i comandi nel seguente formato:

```
verb [param]*
```

dove **verb** è uno dei verbi di amministrazione elencati in [Tabella 38](#) a pagina 716. Tutti i comandi validi contengono un verbo, che viene visualizzato all'inizio del comando in formato standard o breve. I nomi dei verbi non sono sensibili al maiuscolo / minuscolo.

- Per terminare un comando, premere Invio, a meno che non si desideri immettere più comandi insieme, nel qual caso immettere il segno più (+) direttamente prima di premere Invio.

In genere, per terminare i comandi, premere Invio. Tuttavia, è possibile sovrascriverlo immettendo il segno più (+) direttamente prima di premere Invio. Ciò consente di immettere comandi a più righe, come mostrato nel seguente esempio:

```
DEFINE Q(BookingsInputQueue) +
QMGR(QM.POLARIS.TEST) +
QUEUE(BOOKINGS.INPUT.QUEUE) +
PORT(1415) +
CCSID(437)
```

- Per chiudere lo strumento di amministrazione, utilizzare il comando **END**. Questo verbo non può prendere alcun parametro.

Configurazione dei contesti secondari

È possibile utilizzare i verbi **CHANGE**, **DEFINE**, **DISPLAY** e **DELETE** per configurare i contesti secondari dello spazio dei nomi della directory.

Informazioni su questa attività

L'uso di questi verbi è descritto nella tabella seguente.

Tabella 39. Sintassi e descrizione dei comandi utilizzati per manipolare i contesti secondari

Sintassi dei comandi	Descrizione
DEFINE CTX (ctxName)	Tenta di creare un contesto secondario del contesto corrente, con il nome ctxName. Non riesce se si verifica una violazione della sicurezza, se il contesto secondario esiste già o se il nome fornito non è valido.
VISUALIZZA CTX	Visualizza il contenuto del contesto corrente. Gli oggetti gestiti sono annotati con a, i contesti secondari con [D]. Viene visualizzato anche il tipo Java di ciascun oggetto.
DELETE CTX (ctxName)	Tenta di eliminare il contesto child del contesto corrente con il nome ctxName. Non riesce se il contesto non viene trovato, non è vuoto o se si verifica una violazione della sicurezza.
MODIFICA CTX (ctxName)	<p>Modifica il contesto corrente, in modo che ora faccia riferimento al contesto child con il nome ctxName. È possibile fornire uno dei due valori speciali di ctxName :</p> <p>= ATTIVO passa al parent del contesto corrente</p> <p>= INIT passa direttamente al contesto iniziale</p> <p>Ha esito negativo se il contesto specificato non esiste o se si verifica una violazione della sicurezza.</p>

I nomi degli oggetti JMS o Jakarta Messaging memorizzati in un ambiente LDAP devono essere conformi alle convenzioni di denominazione LDAP. Una di queste convenzioni è che i nomi oggetto e contesto devono includere un prefisso, come cn= (nome comune) o ou= (unità organizzativa). Lo strumento di amministrazione semplifica l'utilizzo dei provider di servizi LDAP consentendo di fare riferimento a nomi di oggetti e di contesto senza un prefisso. Se non si fornisce un prefisso, lo strumento aggiunge automaticamente un prefisso predefinito al nome fornito. Per LDAP, è cn=. Se necessario, è possibile modificare il prefisso predefinito impostando la proprietà **NAME_PREFIX** nel file di configurazione. Per ulteriori informazioni, consultare [“Configurazione degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 713.

Nota: Potrebbe essere necessario configurare il server LDAP per memorizzare gli oggetti Java . Per ulteriori informazioni, consultare la documentazione per il server LDAP.

Creazione di oggetti JMS

Per creare oggetti di destinazione e factory di connessione JMS o Jakarta Messaging e memorizzarli in uno spazio dei nomi JNDI, utilizzare il comando DEFINE . Per memorizzare gli oggetti in un ambiente LDAP, è necessario fornire nomi conformi a determinate convenzioni. Lo strumento di amministrazione consente di rispettare le convenzioni di denominazione LDAP aggiungendo un prefisso predefinito ai nomi oggetto.

Informazioni su questa attività

Il comando DEFINE crea un oggetto gestito con il tipo, il nome e le proprietà specificati. Il nuovo oggetto viene memorizzato nel contesto corrente.

I nomi degli oggetti JMS o Jakarta Messaging memorizzati in un ambiente LDAP devono essere conformi alle convenzioni di denominazione LDAP. Una di queste convenzioni è che i nomi oggetto e contesto devono includere un prefisso, come cn= (nome comune) o ou= (unità organizzativa). Lo strumento di amministrazione semplifica l'utilizzo dei provider di servizi LDAP consentendo di fare riferimento a nomi di oggetti e di contesto senza un prefisso. Se non si fornisce un prefisso, lo strumento aggiunge automaticamente un prefisso predefinito al nome fornito. Per LDAP, è cn=. Se necessario, è possibile modificare il prefisso predefinito impostando la proprietà **NAME_PREFIX** nel file di configurazione. Per

ulteriori informazioni, consultare [“Configurazione degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 713.

Nota: Potrebbe essere necessario configurare il server LDAP per memorizzare gli oggetti Java . Per ulteriori informazioni, consultare la documentazione per il server LDAP.

Procedura

1. Se lo strumento di amministrazione non è già stato avviato, avviarlo come descritto in [“Avvio degli strumenti JMSAdmin e JMS30Admin”](#) a pagina 715.
Viene visualizzato il prompt dei comandi, che indica che lo strumento è pronto ad accettare i comandi di gestione.
2. Assicurarsi che il prompt dei comandi mostri il contesto in cui si desidera creare il nuovo oggetto.
Quando si avvia lo strumento di gestione, il prompt inizialmente viene visualizzato come:

```
InitCtx>
```

Per modificare il contesto corrente, utilizzare il comando CHANGE come descritto in [“Configurazione dei contesti secondari”](#) a pagina 717.

3. Per creare una factory di connessione, una destinazione coda o una destinazione argomento, utilizzare la seguente sintassi del comando:

```
DEFINE TYPE (name) [property]*
```

Vale a dire, immettere il comando DEFINE , seguito da un TYPE (name) riferimento oggetto gestito, seguito da zero o più proprietà (consultare [Proprietà di oggetti IBM MQ classes for JMS](#)).

4. Per creare una factory di connessione, una destinazione coda o una destinazione argomento, utilizzare la seguente sintassi del comando:

```
DEFINE TYPE (name) [property]*
```

5. Per visualizzare l'oggetto appena creato, utilizzare il comando DISPLAY con la seguente sintassi del comando:

```
DISPLAY TYPE (name)
```

Esempio

Il seguente esempio mostra una coda denominata testQueue creata nel contesto iniziale utilizzando il comando DEFINE . Poiché questo oggetto viene memorizzato in un ambiente LDAP, anche se il nome oggetto testQueue non viene immesso con un prefisso, lo strumento ne aggiunge automaticamente uno per garantire la conformità con la convenzione di denominazione LDAP. L'inoltro del comando DISPLAY Q(testQueue) comporta anche l'aggiunta di questo prefisso.

```
JM 3.0
InitCtx> DEFINE Q(testQueue)
InitCtx> DISPLAY CTX
Contents of InitCtx
a cn=testQueue          com.ibm.mq.jakarta.jms.MQQueue
1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

JMS 2.0

```
InitCtx> DEFINE Q(testQueue)
InitCtx> DISPLAY CTX
Contents of InitCtx
a cn=testQueue      com.ibm.mq.jms.MQQueue
1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

Condizioni di errore di esempio che creano un oggetto JMS

Quando si crea un oggetto, possono verificarsi diverse condizioni di errore comuni.

Di seguito sono riportati esempi di queste condizioni di errore:

CipherSpec associato a CipherSuite

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

Proprietà non valida per l'oggetto

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

Tipo non valido per il valore della proprietà

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

Conflitto di proprietà - il client/bindings

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

Conflitto di proprietà - Inizializzazione uscita

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

Il valore della proprietà non è compreso nell'intervallo valido

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

Proprietà sconosciuta

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

Di seguito sono riportati esempi di condizioni di errore che potrebbero verificarsi su Windows quando si ricercano oggetti gestiti da JNDI da un'applicazione JMS .

1. Se si utilizza il provider JNDI WebSphere , com.ibm.websphere.naming.WsnInitialContextFactory, è necessario utilizzare una barra (/) per accedere agli oggetti gestiti definiti nei contesti secondari; ad esempio, jms /MyQueueName. Se si utilizza una barra rovesciata (\), viene generata un'eccezione InvalidName.
2. Se si utilizza il provider JNDI Oracle , com.sun.jndi.fscontext.RefFSContextFactory, è necessario utilizzare una barra rovesciata (\) per accedere agli oggetti gestiti definiti nei contesti secondari; ad esempio, ctx1\\fred. Se si utilizza una barra (/), viene generata unaFoundException NameNot.

Configurazione di oggetti JMS

È possibile utilizzare i comandi ALTER, DEFINE, DISPLAY, DELETE, COPY e MOVE per manipolare gli oggetti amministrati nello spazio dei nomi della directory.

Informazioni su questa attività

Tabella 40 a pagina 721 riepiloga l'utilizzo di questi verbi. Sostituire *TYPE* con la parola chiave che rappresenta l'oggetto gestito richiesto, come descritto in [“Configurazione degli oggetti JMS e Jakarta Messaging utilizzando gli strumenti di amministrazione”](#) a pagina 711.

<i>Tabella 40. Sintassi e descrizione dei comandi utilizzati per manipolare gli oggetti gestiti</i>	
Sintassi dei comandi	Descrizione
ALTER <i>TYPE</i> (nome) [proprietà] *	Tenta di aggiornare le proprietà dell'oggetto gestito con quelle fornite. Ha esito negativo se si verifica una violazione della sicurezza, se non è possibile trovare l'oggetto specificato o se le nuove proprietà fornite non sono valide.
DEFINE <i>TYPE</i> (nome) [proprietà] *	Tenta di creare un oggetto gestito di tipo <i>TYPE</i> con le proprietà fornite e lo memorizza con il nome name nel contesto corrente. Non riesce se si verifica una violazione della sicurezza, se il nome fornito non è valido o se esiste un oggetto con tale nome o se le proprietà fornite non sono valide.
DISPLAY <i>TYPE</i> (nome)	Visualizza le proprietà dell'oggetto amministrato di tipo <i>TYPE</i> , associato al nome name nel contesto corrente. Non riesce se l'oggetto non esiste o se si verifica una violazione della sicurezza.
DELETE <i>TYPE</i> (nome)	Tenta di rimuovere l'oggetto gestito di tipo <i>TYPE</i> , con nome name, dal contesto corrente. Non riesce se l'oggetto non esiste o se si verifica una violazione della sicurezza.
COPY <i>TYPE</i> (nameA) <i>TYPE</i> (nameB)	Crea una copia dell'oggetto gestito di tipo <i>TYPE</i> , con il nome nameA, denominando la copia nameB. Tutto ciò si verifica nell'ambito del contesto corrente. Non riesce se l'oggetto da copiare non esiste, se esiste un oggetto denominato nameB o se si verifica una violazione della sicurezza.
MOVE <i>TYPE</i> (nameA) <i>TYPE</i> (nameB)	Sposta (ridenomina) l'oggetto gestito di tipo <i>TYPE</i> , con il nome nameA, in nameB. Tutto ciò si verifica nell'ambito del contesto corrente. Non riesce se l'oggetto da spostare non esiste, se esiste un oggetto con nome nameB o se si verifica una violazione della sicurezza.

JMS 2.0 Configurazione di risorse JMS 2.0 in WebSphere Application Server

Per configurare le risorse JMS 2.0 in WebSphere Application Server, è possibile utilizzare la console di gestione o i comandi wsadmin.

Prima di iniziare

JM 3.0 Sebbene IBM MQ 9.3 e versioni successive supportino Jakarta Messaging 3.0, WebSphere Application Server non dispone attualmente di un supporto equivalente. Pertanto, in WebSphere Application Server, si configurano risorse Java Message Service 2.0 .

Informazioni su questa attività

Le applicazioni Java Message Service 2.0 generalmente si basano su oggetti configurati esternamente che descrivono il modo in cui l'applicazione si connette al proprio fornitore JMS e le destinazioni a cui accede. Le applicazioni JMS utilizzano Java Naming Directory Interface (JNDI) per accedere ai seguenti tipi di oggetto al runtime:

- Specifiche di attivazione (utilizzate dai server delle applicazioni Java EE)
- Le factory di connessione unificate (con JMS 1.1 e versioni successive, le factory di connessione indipendenti dal dominio (unificate) sono preferite alle factory di connessione code specifiche del dominio e alle factory di connessione argomenti)
- Factory di connessione argomenti (utilizzati dalle applicazioni JMS 1.0)
- Factory di connessione code (utilizzate da applicazioni JMS 1.0)
- Code
- Argomenti

Tramite il provider di messaggistica IBM MQ in WebSphere Application Server, le applicazioni di messaggistica Java Message Service (JMS) possono utilizzare il proprio sistema IBM MQ come provider esterno di risorse di messaggistica JMS . Per abilitare questo approccio, si configura il fornitore di messaggistica IBM MQ in WebSphere Application Server per definire le risorse JMS per la connessione a qualsiasi gestore code sulla rete IBM MQ .

È possibile utilizzare WebSphere Application Server per configurare le risorse IBM MQ per le applicazioni (ad esempio factory di connessione code) e per gestire i messaggi e le sottoscrizioni associate alle destinazioni JMS . La sicurezza viene gestita tramite IBM MQ.

Attività correlate

[Utilizzo congiunto di IBM MQ e WebSphere Application Server](#)

WebSphere Application Server argomenti

[Interoperabilità utilizzando il provider di messaggistica IBM MQ](#)

[Gestione della messaggistica con il provider di messaggistica IBM MQ](#)

[Associazione dei nomi del pannello della console di gestione ai nomi di comandi e ai nomi IBM MQ](#)

JMS 2.0 Configurazione delle risorse JMS 2.0 utilizzando la console di gestione

È possibile utilizzare la console di gestione WebSphere Application Server per configurare le specifiche di attivazione, le factory di connessione e le destinazioni per il provider IBM MQ JMS .

Informazioni su questa attività

È possibile utilizzare la console di amministrazione WebSphere Application Server per creare, visualizzare o modificare una delle seguenti risorse:

- Specifiche di attivazione
- Factory di connessione indipendenti dal dominio (JMS 1.1 o versioni successive)
- Factory di connessione code
- Factory di connessione argomenti
- Code
- Argomenti

La seguente procedura fornisce una panoramica dei modi in cui è possibile utilizzare la console di gestione per configurare le risorse JMS da utilizzare con il provider di messaggistica IBM MQ . Ogni passo include il nome dell'argomento nella documentazione del prodotto WebSphere Application Server a cui è possibile fare riferimento per ulteriori informazioni. Consultare *Link correlati* per i link a questi argomenti in IBM Documentation.

In una cella WebSphere Application Server a versione mista, è possibile gestire le risorse IBM MQ sui nodi di tutte le versioni. Tuttavia, alcune proprietà non sono disponibili su tutte le versioni. In questa situazione, nella console di gestione vengono visualizzate solo le proprietà di quel particolare nodo.

Procedura

Per creare o configurare una specifica di attivazione da utilizzare con il fornitore di messaggistica IBM MQ :

- Per creare una specifica di attivazione, utilizzare la procedura guidata Crea risorsa IBM MQ JMS . È possibile utilizzare la procedura guidata per specificare tutti i dettagli per la specifica di attivazione oppure è possibile scegliere di specificare i dettagli di connessione per IBM MQ utilizzando una CCDT (client channel definition table). Quando si specificano i dettagli della connessione utilizzando la procedura guidata, è possibile scegliere di immettere le informazioni sull'host e sulla porta separatamente oppure, se si utilizza un gestore code a più istanze, immettere le informazioni sull'host e sulla porta sotto forma di un elenco di nomi di connessione. Per ulteriori informazioni, consultare *Creazione di una specifica di attivazione per il provider di messaggistica IBM MQ*.
- Per visualizzare o modificare le proprietà di configurazione di una specifica di attivazione, utilizzare il pannello delle impostazioni della factory di connessione del provider di messaggistica IBM MQ della console di gestione.

Queste proprietà di configurazione controllano il modo in cui vengono create le connessioni alle code e agli argomenti associati. Per ulteriori informazioni, fare riferimento a *Configurazione di una specifica di attivazione per il provider di messaggistica IBM MQ*.

Per creare o configurare una factory di connessione unificata, una factory di connessione code o una factory di connessione argomenti da utilizzare con il fornitore di messaggistica IBM MQ :

- Per creare una factory di connessione, selezionare il tipo di factory di connessione che si desidera creare, quindi utilizzare la procedura guidata Crea risorsa IBM MQ JMS per specificare i dettagli.
 - Se l'applicazione JMS è progettata per utilizzare solo la messaggistica point - to - point, creare una factory di connessione specifica del dominio per il dominio di messaggistica point - to - point che può essere utilizzato per creare connessioni specifiche per la messaggistica point - to - point.
 - Se l'applicazione JMS è progettata solo per utilizzare la messaggistica di pubblicazione / sottoscrizione, creare una factory di connessione specifica del dominio per il dominio di messaggistica di pubblicazione / sottoscrizione che può essere utilizzato per la creazione di connessioni specifiche per la messaggistica di pubblicazione / sottoscrizione.
 - Per JMS 1.1 o versioni successive, creare un factory di connessione indipendente dal dominio che può essere utilizzato sia per la messaggistica point-to-point che per la messaggistica di pubblicazione / sottoscrizione, consentendo all'applicazione di eseguire entrambe le operazioni point-to-point e di pubblicazione / sottoscrizione nella stessa transazione.

È possibile scegliere se utilizzare la procedura guidata per specificare tutti i dettagli per la factory di connessione oppure è possibile scegliere di specificare i dettagli di connessione per IBM MQ utilizzando una CCDT (client channel definition table). Quando si specificano i dettagli della connessione utilizzando la procedura guidata, è possibile scegliere di immettere le informazioni sull'host e sulla porta separatamente oppure, se si utilizza un gestore code a più istanze, immettere le informazioni sull'host e sulla porta sotto forma di un elenco di nomi di connessione. Per ulteriori informazioni, consultare *Creazione di una factory di connessione per il provider dei messaggi IBM MQ*.

Per visualizzare o modificare le proprietà di configurazione di un factory di connessione:

- Utilizzare il pannello delle impostazioni della produzione connessioni della console di gestione per il tipo di produzione connessioni che si desidera configurare.

Le proprietà di configurazione controllano il modo in cui vengono create le connessioni alle code e agli argomenti associati. Per ulteriori informazioni, consultare *Configurazione di una factory di raccolta per il provider di messaggistica IBM MQ*, *Configurazione di una factory di raccolta code per il provider di messaggistica IBM MQ* o *Configurazione di una factory di raccolta argomenti per il provider di messaggistica IBM MQ*.

Per configurare una destinazione coda JMS per la messaggistica point - to - point con il fornitore di messaggistica IBM MQ :

- Utilizzare il pannello delle impostazioni della coda del provider di messaggistica IBM MQ della console di gestione per definire i seguenti tipi di proprietà:
 - Proprietà generali, incluse le proprietà di gestione e delle code IBM MQ .
 - Proprietà di connessione che specifica come connettersi al gestore code su cui è presente la coda.
 - Proprietà avanzate che controllano il funzionamento delle connessioni effettuate alle destinazioni dei provider dei messaggi IBM MQ .
 - Qualsiasi proprietà personalizzata per la destinazione della coda.

Per ulteriori informazioni, consultare *Configurazione di una coda per il provider di messaggistica IBM MQ*.

Per creare o configurare una destinazione argomenti JMS per la messaggistica di pubblicazione / sottoscrizione con il provider di messaggistica IBM MQ :

- Utilizzare il pannello di impostazioni dell'argomento del fornitore di messaggistica IBM MQ per definire i tipi di proprietà seguenti:
 - Proprietà generali, incluse le proprietà di amministrazione e IBM MQ .
 - Proprietà avanzate che controllano il funzionamento delle connessioni effettuate alle destinazioni dei provider dei messaggi IBM MQ .
 - Qualsiasi proprietà personalizzata per la destinazione della coda.

Per ulteriori informazioni, consultare *Configurazione di un argomento per il provider di messaggistica IBM MQ*.

Concetti correlati

[“Gestori code a più istanze” a pagina 524](#)

I gestori code a più istanze sono istanze dello stesso gestore code configurato su server differenti. Un'istanza del gestore code è definita come istanza attiva e un'istanza è definita come istanza in standby. Se l'istanza attiva ha esito negativo, il gestore code a più istanze viene riavviato automaticamente sul server di standby.

Attività correlate

[“Configurazione di un formato binario CCDT” a pagina 44](#)

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

[“Configurazione della messaggistica di pubblicazione/sottoscrizione” a pagina 448](#)

È possibile avviare, arrestare e visualizzare lo stato della pubblicazione / sottoscrizione in coda. È inoltre possibile aggiungere e rimuovere i flussi e aggiungere ed eliminare i gestori code da una gerarchia broker.

WebSphere Application Server argomenti

[Specifiche di attivazione del fornitore di messaggistica IBM MQ](#)

[Creazione di una specifica di attivazione per il provider di messaggistica IBM MQ](#)

[Configurazione di una specifica di attivazione per il provider di messaggistica IBM MQ](#)

[Creazione di una factory di connessione per il provider dei messaggi IBM MQ](#)

[Configurazione di una factory di connessione unificata per il provider di messaggistica IBM MQ](#)

[Configurazione di una factory di connessione code per il provider di messaggistica IBM MQ](#)

[Configurazione di una factory di connessione argomenti per il provider dei messaggi IBM MQ](#)

JMS 2.0 Configurazione delle risorse JMS 2.0 utilizzando i comandi di script wsadmin

È possibile utilizzare i comandi di script WebSphere Application Server wsadmin per creare, modificare, eliminare o visualizzare informazioni sulle specifiche di attivazione JMS , sulle factory di connessione, sulle code e sugli argomenti. È inoltre possibile visualizzare e gestire le impostazioni per l'adattatore risorse IBM MQ .

Informazioni su questa attività

I seguenti passi forniscono una panoramica dei modi in cui è possibile utilizzare comandi WebSphere Application Server wsadmin per configurare le risorse JMS da utilizzare con il provider di messaggistica IBM MQ . Per ulteriori informazioni su come utilizzare questi comandi, consultare *Link correlati* per i link alla documentazione del prodotto WebSphere Application Server .

Per eseguire un comando, utilizzare l'oggetto AdminTask del client di script wsadmin.

Dopo aver utilizzato un comando per creare un nuovo oggetto o apportare modifiche, salvare le proprie modifiche alla configurazione principale. Ad esempio, utilizza il seguente comando:

```
AdminConfig.save()
```

Per visualizzare un elenco dei comandi di gestione del provider dei messaggi IBM MQ disponibili e una breve descrizione di ciascun comando, immettere il seguente comando dal prompt wsadmin:

```
print AdminTask.help('WMQAdminCommands')
```

Per visualizzare la guida di panoramica su un determinato comando, immettere il seguente comando nel prompt wsadmin:

```
print AdminTask.help('command_name')
```

Procedura

Per elencare tutte le risorse del provider dei messaggi IBM MQ definite nell'ambito in cui viene eseguito un comando, utilizzare i seguenti comandi.

- Per elencare le specifiche di attivazione, utilizzare il comando **listWMQActivationSpecs** .
- Per elencare le produzioni connessioni, utilizzare il comando **listWMQConnectionFactory** .
- Per elencare le destinazioni di tipo coda, utilizzare il comando **listWMQQueues** .
- Per elencare le destinazioni del tipo di argomento, utilizzare il comando **listWMQTopics** .

Per creare una risorsa JMS per il fornitore di messaggistica IBM MQ in un ambito specifico, utilizzare i seguenti comandi.

- Per creare una specifica di attivazione, utilizzare il comando **createWMQActivationSpec** .
È possibile creare una specifica di attivazione specificando tutti i parametri da utilizzare per stabilire una connessione oppure è possibile creare la specifica di attivazione in modo che utilizzi una CCDT (client channel definition table) per individuare il gestore code a cui connettersi.
- Per creare una factory di connessione, utilizzare il comando **createWMQConnectionFactory** , utilizzando il parametro **-type** per specificare il tipo di factory di connessione che si desidera creare:
 - Se l'applicazione JMS è progettata per utilizzare solo la messaggistica point - to - point, creare una factory di connessione specifica del dominio per il dominio di messaggistica point - to - point che può essere utilizzato per creare connessioni specifiche per la messaggistica point - to - point.

- Se l'applicazione JMS è progettata solo per utilizzare la messaggistica di pubblicazione / sottoscrizione, creare una factory di connessione specifica del dominio per il dominio di messaggistica di pubblicazione / sottoscrizione che può essere utilizzato per la creazione di connessioni specifiche per la messaggistica di pubblicazione / sottoscrizione.
- Per JMS 1.1 o versioni successive, creare un factory di connessione indipendente dal dominio che può essere utilizzato sia per la messaggistica point-to-point che per la messaggistica di pubblicazione / sottoscrizione, consentendo all'applicazione di eseguire entrambe le operazioni point-to-point e di pubblicazione / sottoscrizione nella stessa transazione.

Il tipo predefinito è factory di connessione indipendente dal dominio.

- Per creare una destinazione di tipo coda, utilizzare il comando **createWMQQueue** .
 - Per creare una destinazione del tipo di argomento, utilizzare il comando **createWMQTopic** .
- Per modificare una risorsa JMS per il fornitore di messaggistica IBM MQ in un ambito specifico, utilizzare i seguenti comandi.

- Per modificare una specifica di attivazione, utilizzare il comando **modifyWMQActivationSpec** .
Non è possibile modificare il tipo di una specifica di attivazione. Ad esempio, non è possibile creare una specifica di attivazione in cui si immettono tutte le informazioni di configurazione manualmente e quindi modificarle per utilizzare una CCDT.

- Per modificare una factory di connessione, utilizzare il comando **modifyWMQConnectionFactory** .
- Per modificare una destinazione del tipo di coda, utilizzare il comando **modifyWMQQueue** .
- Per modificare una destinazione del tipo di argomento, utilizzare il comando **modifyWMQTopic** .

Per cancellare una risorsa JMS per il fornitore di messaggistica IBM MQ in un ambito specifico, utilizzare i seguenti comandi.

- Per cancellare una specifica di attivazione, utilizzare il comando **deleteWMQActivationSpec** .
- Per eliminare una factory di connessione, utilizzare il comando **deleteWMQConnectionFactory** .
- Per eliminare una destinazione di tipo coda, utilizzare il comando **deleteWMQQueue** .
- Per cancellare una destinazione di tipo argomento, utilizzare il comando **deleteWMQTopic** .

Per visualizzare informazioni su una specifica risorsa del fornitore di messaggistica IBM MQ , utilizzare i seguenti comandi.

- Per visualizzare tutti i parametri e i relativi valori associati a una particolare specifica di attivazione, utilizzare il comando **showWMQActivationSpec** .
- Per visualizzare tutti i parametri ed i relativi valori associati ad un determinato factory di connessione, utilizzare il comando **showWMQConnectionFactory** .
- Per visualizzare tutti i parametri e i relativi valori associati a una particolare destinazione del tipo di coda, utilizzare il comando **showWMQQueue** .
- Per visualizzare tutti i parametri e i relativi valori associati a una destinazione di tipo argomento, utilizzare il comando **deleteWMQTopic** .

Per gestire le impostazioni per l'adattatore di risorse IBM MQ o per il provider di messaggistica IBM MQ , utilizzare i seguenti comandi.

- Per gestire le impostazioni dell'adattatore di risorse IBM MQ installato in un particolare ambito, utilizzare il comando **manageWMQ** .
- Per visualizzare tutti i parametri e i relativi valori che possono essere impostati dal comando **manageWMQ** , utilizzare il comando **showWMQ** . Queste impostazioni sono correlate all'adattatore di risorse IBM MQ o al fornitore di messaggistica IBM MQ . Il comando **showWMQ** mostra anche le proprietà personalizzate impostate sull'adattatore di risorse IBM MQ .

Concetti correlati

[“Gestori code a più istanze” a pagina 524](#)

I gestori code a più istanze dello stesso gestore code configurato su server differenti.

Un'istanza del gestore code è definita come istanza attiva e un'istanza è definita come istanza in standby.

Se l'istanza attiva ha esito negativo, il gestore code a più istanze viene riavviato automaticamente sul server di standby.

Attività correlate

[“Configurazione di un formato binario CCDT” a pagina 44](#)

La tabella di definizione del canale client (CCDT) determina le definizioni di canale e le informazioni di autenticazione utilizzate dalle applicazioni client per connettersi al gestore code. Su Multiplatforms, una CCDT binaria contenente le impostazioni predefinite viene creata automaticamente quando viene creato il gestore code. Utilizzare il comando **runmqsc** per aggiornare una CCDT binaria.

[“Configurazione della messaggistica di pubblicazione/sottoscrizione” a pagina 448](#)

È possibile avviare, arrestare e visualizzare lo stato della pubblicazione / sottoscrizione in coda. È inoltre possibile aggiungere e rimuovere i flussi e aggiungere ed eliminare i gestori code da una gerarchia broker.

WebSphere Application Server argomenti

[**createWMQActivationSpec** comando](#)

[**createWMQConnectionFactory** comando](#)

[**createWMQQueue** comando](#)

[**createWMQTopic** comando](#)

[**deleteWMQActivationSpec** comando](#)

[**deleteWMQConnectionFactory** comando](#)

[**deleteWMQQueue** comando](#)

[**deleteWMQTopic** comando](#)

[**listWMQActivationSpecs** comando](#)

[**listWMQConnectionFactories** comando](#)

[**listWMQQueues** comando](#)

[**listWMQTopics** comando](#)

[**modifyWMQActivationSpec** comando](#)

[**modifyWMQConnectionFactory** comando](#)

[**modifyWMQQueue** comando](#)

[**modifyWMQTopic** comando](#)

[**showWMQActivationSpec** comando](#)

[**showWMQConnectionFactory** comando](#)

[**showWMQQueue** comando](#)

[**showWMQTopic** comando](#)

[**showWMQ** comando](#)

[**manageWMQ** comando](#)

JMS 2.0 Utilizzo delle sottoscrizioni condivise JMS 2.0

In WebSphere Application Server traditional 9.0, puoi configurare e utilizzare le sottoscrizioni condivise JMS 2.0 con IBM MQ 9.0.

Informazioni su questa attività

La specificazione JMS 2.0 ha introdotto il concetto di sottoscrizioni condivise, che consente a una singola sottoscrizione di essere aperta da uno o più consumer. I messaggi sono condivisi tra tutti questi consumatori. Non vi è alcuna limitazione quando questi consumer sono così lunghi che si connettono allo stesso gestore code.

Le sottoscrizioni condivise possono essere durevoli o non durevoli, con la stessa semantica di quelle a cui ora si fa riferimento come sottoscrizioni non condivise.

Per consentire a un utente di identificare quale sottoscrizione utilizzare, è necessario fornire un nome sottoscrizione. È simile alle sottoscrizioni durevoli non condivise, ma è richiesto un nome di sottoscrizione in tutti i casi in cui è richiesta una sottoscrizione condivisa. Un clientID, tuttavia, non è richiesto nel caso di una sottoscrizione condivisa durevole; può essere fornito ma non è obbligatorio.

Mentre le sottoscrizioni condivise possono essere considerate come un meccanismo di bilanciamento del carico, né in IBM MQ né nella specifica JMS 2.0 vi è alcun impegno su come i messaggi vengono distribuiti tra i consumatori.

In WebSphere Application Server traditional 9.0 è preinstallato un adattatore di risorse IBM MQ 9.0 .

La seguente procedura mostra come configurare una specifica di attivazione per utilizzare una sottoscrizione condivisa durevole o non durevole utilizzando la console di gestione WebSphere Application Server traditional .

Procedura

Creare prima gli oggetti in JNDI.

1. Creare una destinazione argomenti in JNDI come di consueto (consultare [“Configurazione delle risorse JMS 2.0 utilizzando la console di gestione”](#) a pagina 722).
2. Creare la specifica di attivazione (consultare [“Configurazione delle risorse JMS 2.0 utilizzando la console di gestione”](#) a pagina 722).

È possibile creare la specifica di attivazione con esattamente le proprietà necessarie. Se si desidera utilizzare una sottoscrizione durevole, è possibile selezionarla al momento della creazione e specificare un nome. Se si desidera utilizzare una sottoscrizione non durevole, non è possibile specificare un nome a questo punto. Invece, è necessario creare una proprietà personalizzata per il nome della sottoscrizione.

Aggiornare la specifica di attivazione creata con le proprietà personalizzate richieste. Potrebbero essere necessarie due proprietà personalizzate:

- In tutti i casi, è necessario creare una proprietà personalizzata per specificare che questa specifica di attivazione deve utilizzare una sottoscrizione condivisa.
- Se la sottoscrizione è stata creata come non durevole, la proprietà del nome della sottoscrizione deve essere impostata come proprietà personalizzata.

La seguente tabella mostra il valore valido che è possibile specificare per ciascuna proprietà personalizzata:

Nome proprietà	Tipo	Valori validi
sharedSubscription	Stringa	true, false
subscriptionName	Stringa	Stringa java di lunghezza diversa da zero

3. Selezionare la specifica di attivazione dall'elenco visualizzato nel modulo **Raccolta specifica di attivazione** .

I dettagli per la specifica di attivazione vengono visualizzati nel modulo **Impostazioni della specifica di attivazione del fornitore di messaggistica IBM MQ** .

4. Nel modulo **IBM MQ** , fare clic su **Proprietà personalizzate**.

Viene visualizzato il modulo **Proprietà personalizzate** .

5. Se si utilizza una sottoscrizione non durevole, creare la proprietà personalizzata subscriptionName .

Sul pannello **Proprietà personalizzate** della specifica di attivazione, fare clic su **Nuovo**, quindi immettere i seguenti dettagli:

Nome

Il nome della proprietà personalizzata, che in questo caso è subscriptionName.

Valore

Il valore della proprietà personalizzata. È possibile utilizzare i nomi JNDI nel campo **Valore** , ad esempio WASSharedSubOne.

Tipo

Il tipo di proprietà personalizzata. Selezionare il tipo di proprietà personalizzata dall'elenco, che in questo caso deve essere `java.lang.String`.

6. Per una sottoscrizione durevole e non durevole condivisa, creare la proprietà personalizzata `sharedSubscription`.

Sul pannello **Proprietà personalizzate** della specifica di attivazione, fare clic su **Nuovo**, quindi immettere i seguenti dettagli:

Nome

Il nome della proprietà personalizzata, che in questo caso è `sharedSubscription`.

Valore

Il valore della proprietà personalizzata. Per specificare che la specifica di attivazione utilizza una sottoscrizione condivisa, impostare il valore su `true`. Se in seguito si desidera interrompere l'utilizzo di una sottoscrizione condivisa per questa specifica di attivazione, è possibile farlo impostando il valore di questa proprietà personalizzata su `false`.

Tipo

Il tipo di proprietà personalizzata. Selezionare il tipo di proprietà personalizzata dall'elenco, che in questo caso deve essere `java.lang.String`.

7. Quando le proprietà sono impostate, riavviare il server di applicazioni.

I bean basati sui messaggi (MDB) per le specifiche di attivazione vengono guidati quando arrivano i messaggi, ma solo gli MDB condividono i messaggi inviati.

Concetti correlati

[Sottoscrizioni clonate e condivise](#)

[Durata sottoscrizione](#)

Attività correlate

[Configurazione dell'adattatore di risorse per la comunicazione in entrata](#)

Informazioni correlate per WebSphere Application Server traditional 9.0

[Configurazione di un argomento per il provider di messaggistica IBM MQ](#)

[Specifiche di attivazione del fornitore di messaggistica IBM MQ](#)

[Creazione di una specifica di attivazione per il provider di messaggistica IBM MQ](#)

[Configurazione di una specifica di attivazione per il provider di messaggistica IBM MQ](#)

[Configurazione delle proprietà personalizzate per le risorse del IBM MQ provider di messaggistica JMS](#)

JMS 2.0

Utilizzo delle proprietà JMS 2.0 ConnectionFactory e Destination

Lookup

In WebSphere Application Server traditional 9.0, le proprietà `ConnectionFactoryLookup` e `DestinationLookup` di una specifica di attivazione possono essere fornite con un nome JNDI di un oggetto gestito da utilizzare come preferenza rispetto alle altre proprietà della specifica di attivazione.

Informazioni su questa attività

La specifica JMS 2.0 specifica due proprietà aggiuntive sulla specifica di attivazione utilizzata per guidare MDB (message - driven bean). In precedenza, ogni fornitore doveva specificare le proprietà personalizzate nella specifica di attivazione per fornire i dettagli richiesti per la connessione a un sistema di messaggistica e per definire la destinazione da cui ottenere i messaggi.

Le proprietà `connectionFactoryLookup` e `destinationLookup` ora standard possono essere utilizzate per fornire un JNDI nome dell'oggetto pertinente da cercare e utilizzare. All'interno di WebSphere Application Server traditional 9.0 un adattatore risorse IBM MQ 9.0 è preinstallato.

La seguente procedura mostra come personalizzare e utilizzare queste due proprietà utilizzando la console di gestione WebSphere Application Server traditional .

Procedura

Creare prima gli oggetti in JNDI.

1. Creare ConnectionFactory in JNDI come di consueto (consultare [“Configurazione delle risorse JMS 2.0 utilizzando la console di gestione”](#) a pagina 722).
2. Creare la destinazione in JNDI come normale (consultare [“Configurazione delle risorse JMS 2.0 utilizzando la console di gestione”](#) a pagina 722).

L'oggetto Destinazione deve avere i valori corretti.

3. Creare la specifica di attivazione utilizzando i valori necessari (consultare [“Configurazione delle risorse JMS 2.0 utilizzando la console di gestione”](#) a pagina 722).

È possibile creare la specifica di attivazione con esattamente le proprietà necessarie. Tuttavia, è necessario tenere presente le seguenti considerazioni:

- Se si desidera che l'adattatore di risorse IBM MQ utilizzi le proprietà di ricerca della destinazione e della factory di connessione Java EE , è meno rilevante quali proprietà vengono utilizzate quando si crea la specifica di attivazione (consultare [ActivationSpec ConnectionFactoryLookup e DestinationLookup properties](#)).
- Tuttavia, qualsiasi proprietà che non sia già definita sul factory di connessione o sulla destinazione deve essere ancora specificata sulla specifica di attivazione. Pertanto, è necessario definire le proprietà del consumer della connessione e le proprietà aggiuntive e le informazioni di autenticazione utilizzate quando viene effettivamente creata una connessione.
- Delle proprietà definite sul factory di connessione, la proprietà ClientID ha un'elaborazione speciale. Ciò si verifica perché uno scenario comune utilizza un singolo factory di connessione con più specifiche di attivazione. Ciò semplifica la gestione, tuttavia la specifica JMS richiede ID client univoci, pertanto la specifica di attivazione deve avere la possibilità di sovrascrivere qualsiasi valore impostato in ConnectionFactory. Se non viene impostato alcun ClientID sulla specifica di attivazione, viene utilizzato qualsiasi valore sul factory di connessione.

Aggiornare la specifica di attivazione creata con le due nuove proprietà personalizzate utilizzando la console di gestione WebSphere Application Server come descritto nel passo [“4”](#) a pagina 730 oppure utilizzare le annotazioni come descritto nel passo [“5”](#) a pagina 731.

4. Aggiornare la specifica di attivazione nella console di gestione WebSphere Application Server .

Queste due proprietà devono essere impostate nel pannello delle proprietà personalizzate della specifica di attivazione. Queste proprietà non sono presenti nei pannelli principali della specifica di attivazione o nella procedura guidata di creazione della specifica di attivazione.

- a) Selezionare la specifica di attivazione dall'elenco visualizzato nel modulo **Raccolta specifica di attivazione** .

I dettagli per la specifica di attivazione vengono visualizzati nel modulo **Impostazioni della specifica di attivazione del fornitore di messaggistica IBM MQ** .

- b) Nel modulo **IBM MQ** , fare clic su **Proprietà personalizzate**.

Viene visualizzato il modulo **Proprietà personalizzate** .

- c) Nel modulo **Proprietà personalizzate** , creare due nuove proprietà personalizzate, di tipo java.lang.String.

In ogni caso, fare clic su **Nuovo** , quindi immettere i dettagli seguenti per la proprietà personalizzata:

Nome

Il nome della proprietà personalizzata, connectionFactoryLookup o destinationLookup.

Valore

Il valore della proprietà personalizzata. È possibile utilizzare i nomi JNDI nel campo **Valore** , ad esempio QuoteCF e QuoteQ.

Tipo

Il tipo di proprietà personalizzata. Selezionare il tipo di proprietà personalizzata dall'elenco, che in questo caso deve essere `java.lang.String`.

L'MDB distribuito utilizzerà questi valori per creare la factory di connessione e la destinazione. Quando si distribuisce l'MDB, non è necessario impostare la configurazione del valore JNDI .

5. Utilizzare le annotazioni invece della specifica di attivazione.

È possibile utilizzare le annotazioni nel codice MDB per specificare anche i valori. Ad esempio, utilizzando JNDI names `QuoteCF` e `QuoteQ`, il codice è simile al seguente:

```
@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationType" , propertyValue =
"javax.jms.Topic" ),
    @ActivationConfigProperty(propertyName = "destinationLookup" , propertyValue =
"QuoteQ" ),
    @ActivationConfigProperty(propertyName = "connectionFactoryLookup" , propertyValue
= "QuoteCF" )}, mappedName = "LookupMDB" )
@TransactionAttribute(TransactionAttributeType.REQUIRED)
@TransactionManagement(TransactionManagementType.CONTAINER)
publicclass LookupMDB implements MessageListener {
```

Attività correlate

[Configurazione dell'adattatore di risorse per la comunicazione in entrata](#)

Informazioni correlate per WebSphere Application Server traditional 9.0

[Configurazione di una factory di connessione unificata per il provider di messaggistica IBM MQ](#)

[Configurazione di un argomento per il provider di messaggistica IBM MQ](#)

[Specifiche di attivazione del fornitore di messaggistica IBM MQ](#)

[Creazione di una specifica di attivazione per il provider di messaggistica IBM MQ](#)

[Configurazione di una specifica di attivazione per il provider di messaggistica IBM MQ](#)

[Configurazione delle proprietà personalizzate per le risorse del IBM MQ provider di messaggistica JMS](#)

Configurazione di WebSphere Application Server per utilizzare il livello di manutenzione dell'adattatore di risorse più recente

Per garantire che l'adattatore di risorse IBM MQ venga aggiornato automaticamente all'ultimo livello di manutenzione disponibile quando si applicano i fix pack WebSphere Application Server , è possibile configurare tutti i server nel proprio ambiente per utilizzare la versione più recente dell'adattatore di risorse contenuto nel fix pack WebSphere Application Server applicato all'installazione di ciascun nodo.

Prima di iniziare

Importante:

- **JM 3.0** WebSphere Application Server traditional non supporta attualmente Jakarta EE. Consultare [IBM MQ resource adapter statement of support](#).
- Se si sta utilizzando WebSphere Application Server 8.5 o versioni precedenti su qualsiasi piattaforma, non installare l'adattatore di risorse IBM MQ 8.0 o versioni successive nel server delle applicazioni. L'adattatore di risorse IBM MQ 8.0 o successivo può essere distribuito solo in un server delle applicazioni che supporta JMS 2.0. Tuttavia, WebSphere Application Server 8.5 o versioni precedenti supporta solo JMS 1.1.

Informazioni su questa attività

Utilizzare questa attività se una delle seguenti circostanze si applica alla configurazione e si desidera configurare tutti i server nel proprio ambiente per utilizzare la versione più recente dell'adattatore di risorse IBM MQ :

- I log JVM di qualsiasi server delle applicazioni nel proprio ambiente mostrano le seguenti informazioni sulla versione dell'adattatore di risorse IBM MQ dopo l'applicazione di WebSphere Application Server 7.0.0 Fix Pack 1 o successive:

```
WMSG1703I:RAR versione di implementazione 7.0.0.0-k700-L080820
```

- I log JVM di qualsiasi server delle applicazioni nel proprio ambiente contengono la seguente voce:


```
WMSG1625E: Non è stato possibile rilevare
il codice provider di messaggistica IBM MQ nel percorso specificato < null>
```
- Uno o più nodi sono stati precedentemente aggiornati manualmente per utilizzare un livello di manutenzione specifico dell'adattatore di risorse IBM MQ che è ora sostituito dalla versione più recente dell'adattatore di risorse contenuto nel livello di manutenzione WebSphere Application Server corrente.

La directory *profile_root* a cui fanno riferimento gli esempi è la directory home per il profilo WebSphere Application Server , ad esempio C:\Program Files\IBM\WebSphere\AppServer1.

Una volta eseguita la seguente procedura per tutte le celle e le installazioni di un singolo server nel proprio ambiente, i server ricevono automaticamente la manutenzione per l'adattatore di risorse IBM MQ quando viene applicato un nuovo fix pack WebSphere Application Server .

Procedura

1. Avviare il server delle applicazioni. Se il profilo fa parte di una configurazione di distribuzione di rete, avviare il gestore distribuzione e tutti gli agent del nodo. Se il profilo contiene un agent di gestione, avviare l'agent di gestione.
2. Verificare il livello di manutenzione dell'adattatore di risorse IBM MQ .
 - a) Aprire una finestra del prompt dei comandi e passare alla directory *profile_root*\bin .
Ad esempio, immettere `cd C:\Program Files\IBM\WebSphere\AppServer1\bin`.
 - b) Avviare lo strumento wsadmin immettendo `wsadmin.bat -lang jython`, quindi, se richiesto, immettere il nome utente e la password.
 - c) Immettere il comando riportato di seguito e premere due volte Invio:

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WMQInfo,*")
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean,
'getInfo', '')
```

È possibile eseguire questo comando anche in Jacl. Per ulteriori informazioni su come effettuare questa operazione, consultare *Verifica che i server utilizzino il livello di manutenzione dell'adattatore di risorse IBM MQ disponibile più recente* nella documentazione del prodotto WebSphere Application Server .

- d) Individuare il messaggio WMSG1703I nell'emissione visualizzata dal comando e controllare il livello dell'adattatore risorse.

Ad esempio, per WebSphere Application Server 7.0.1 Fix Pack 5, il messaggio dovrebbe essere:

```
WMSG1703I: Versione implementazione RAR 7.0.1.3-k701-103-100812
```

Questo messaggio indica che la versione è 7.0.1.3-k701-103-100812, che è il corretto livello dell'adattatore di risorse per questo fix pack. Tuttavia, se invece viene visualizzato il seguente messaggio, ciò significa che è necessario adattare l'adattatore di risorse al livello di manutenzione corretto per WebSphere Application Server 7.0.1 Fix Pack 5.

```
WMSG1703I: Versione implementazione RAR 7.0.0.0-k700-L080820
```

3. Copiare il seguente script Jython in un file denominato `convertWMQRA.py`, quindi salvarlo nella directory root del profilo, ad esempio C:\Program Files\IBM\WebSphere\AppServer1\bin.

```
ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
    desc = AdminConfig.showAttribute(ra, "description")
    if (desc == "WAS 7.0 Built In MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In MQ Resource
Adapter"):
        print "Updating archivePath and classpath of " + ra
        AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
```

```
AdminConfig.unsetAttribute(ra, ['classpath'])
AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
AdminConfig.save()
#end if
#end for
```

Suggerimento: Quando si salva il file, assicurarsi che sia salvato come file python piuttosto che come file di testo.

4. Utilizzare lo strumento wsadmin WebSphere Application Server per eseguire lo script Jython appena creato.

Aprire un prompt dei comandi e passare alla directory `\bin` nella directory home per la directory WebSphere Application Server, ad esempio `C:\Program Files\IBM\WebSphere\AppServer1\bin`, quindi immettere il seguente comando e premere Invio:

```
wsadmin -lang jython -f convertWMQRA.py
```

Se richiesto, immettere il nome utente e la password.

Nota: Se si esegue lo script su un profilo che fa parte di una configurazione di distribuzione di rete, lo script aggiorna tutti i profili che devono essere aggiornati in tale configurazione. Una risincronizzazione completa potrebbe essere necessaria se si hanno incongruenze del file di configurazione preesistente.

5. Se si è in esecuzione in una configurazione di distribuzione di rete, assicurarsi che gli agent del nodo siano completamente risincronizzati. Per ulteriori informazioni, consultare la sezione Sincronizzazione dei nodi mediante lo strumento di script wsadmin o Aggiunta, gestione e rimozione di nodi.
6. Arrestare tutti i server nel profilo. Se il profilo fa parte di una configurazione di distribuzione di rete, arrestare anche tutti i membri del cluster nella configurazione, arrestare tutti gli agent del nodo nella configurazione e arrestare il gestore distribuzione. Se il profilo contiene un agent di gestione, arrestarlo.
7. Eseguire il comando **osgiCfgInit** dalla directory `profile_root/bin`.
Il comando `osgiCfgInit` reimposta la cache delle classi utilizzata dall'ambiente di runtime OSGi. Se il profilo fa parte di una configurazione di distribuzione di rete, eseguire il comando **osgiCfgInit** dalla directory `profile_root/bin` di ogni profilo che è parte della configurazione.
8. Riavviare tutti i server nel profilo. Se il profilo fa parte di una configurazione di distribuzione di rete, riavviare anche tutti i membri del cluster nella configurazione, riavviare tutti gli agent del nodo nella configurazione e riavviare il gestore distribuzione. Se il profilo contiene un agent di gestione, riavviarlo.
9. Ripetere il passo 2 per controllare che l'adattatore risorse sia ora al livello corretto.

Operazioni successive

Se si continuano a riscontrare dei problemi dopo aver eseguito i passi descritti in questo argomento ed è stato precedentemente utilizzato il pulsante **Aggiorna adattatore risorse** nel pannello JMS Impostazioni provider nella console di gestione WebSphere Application Server per aggiornare l'adattatore risorse IBM MQ su qualsiasi nodo nel proprio ambiente, è possibile che si stia verificando il problema descritto in [APAR PM10308](#).

Concetti correlati

[Utilizzo dell'adattatore di risorse IBM MQ](#)

Informazioni correlate per WebSphere Application Server 8.5.5

[Verifica che i server utilizzino il livello di manutenzione dell'adattatore di risorse IBM MQ più recente disponibile](#)

[Sincronizzazione dei nodi mediante lo strumento di script wsadmin](#)

[Aggiunta, gestione e rimozione di nodi](#)

[JMS Impostazioni provider](#)

Configurazione della proprietà JMS PROVIDERVERSION

Il provider di messaggistica IBM MQ ha tre modalità operative: modalità normale, modalità normale con limitazioni e modalità di migrazione. È possibile impostare la proprietà JMS **PROVIDERVERSION** per selezionare quali di queste modalità un'applicazione JMS utilizza per pubblicare e sottoscrivere.

Informazioni su questa attività

La selezione della modalità operativa del provider di messaggistica IBM MQ può essere controllata principalmente impostando la proprietà della factory di connessione PROVIDERVERSION. La modalità di funzionamento può essere selezionata automaticamente anche se non è stata specificata una modalità.

La proprietà **PROVIDERVERSION** differenzia tra le tre modalità operative del provider di messaggistica IBM MQ :

Modalità normale del provider di messaggistica IBM MQ

La modalità normale utilizza tutte le funzioni di un gestore code IBM MQ per implementare JMS. Questa modalità è ottimizzata per utilizzare la API e la funzionalità JMS 2.0.

Modalità normale del provider di messaggistica IBM MQ con restrizioni

La modalità normale con limitazioni utilizza l'API JMS 2.0 , ma non le nuove funzioni, ossia le sottoscrizioni condivise, la consegna ritardata e l'invio asincrono.

Modalità di migrazione del provider di messaggistica IBM MQ

Con la modalità di migrazione, è possibile connettersi a un gestore code IBM MQ 8.0 o successivo, ma non viene utilizzata alcuna delle funzioni di un gestore code IBM WebSphere MQ 7.0 o successivo, come la lettura anticipata e lo streaming.

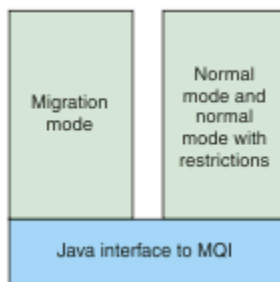


Figura 89. Modalità del provider di messaggistica

Procedura

Per configurare la proprietà **PROVIDERVERSION** per una specifica factory di connessione:

- Per configurare la proprietà **PROVIDERVERSION** utilizzando IBM MQ Explorer, fare riferimento a [Configurazione di gestori code e oggetti](#).
- Per configurare la proprietà **PROVIDERVERSION** utilizzando lo strumento di gestione JMS , fare riferimento a [Configurazione di gestori code e oggetti](#).
- Per configurare la proprietà **PROVIDERVERSION** in un'applicazione JMS utilizzando le estensioni IBM JMS o IBM MQ JMS , consultare [Creazione e configurazione di factory di connessione e destinazioni in un'applicazione IBM MQ classes for JMS](#).

Per sovrascrivere le impostazioni della modalità del provider del factory di connessione per tutte le factory di connessione nella JVM:

- Per sovrascrivere le impostazioni della modalità del provider del factory di connessione, utilizzare la proprietà `com.ibm.msg.client.wmq.overrideProviderVersion`

Se non è possibile modificare la factory di connessione che si sta utilizzando, è possibile utilizzare la proprietà `com.ibm.msg.client.wmq.overrideProviderVersion` per sovrascrivere qualsiasi impostazione sulla factory di connessione. Questa sovrascrittura si applica a tutte le factory di connessione della JVM, ma gli oggetti factory di connessione effettivi non vengono modificati.

Concetti correlati

Risoluzione dei problemi relativi alla versione del provider JMS

Riferimenti correlati

[PROVIDERVERSION](#)

[Proprietà delle factory di connessione](#)

[Dipendenze tra proprietà di oggetti IBM MQ classes for JMS](#)

Modalità operative del provider di messaggistica IBM MQ

È possibile selezionare la modalità operativa del provider di messaggistica IBM MQ che un'applicazione JMS utilizza per pubblicare e sottoscrivere impostando la proprietà PROVIDERVERSION per la factory di connessione sul valore appropriato. In alcuni casi, la proprietà PROVIDERVERSION è impostata come non specificata, nel cui caso il client JMS utilizza un algoritmo per stabilire quale modalità di operazione utilizzare.

PROVIDERVERSION Valori proprietà

È possibile impostare la proprietà **PROVIDERVERSION** del factory di connessione su uno qualsiasi dei seguenti valori:

8 - Modalità normale

L'applicazione JMS utilizza la modalità normale. Questa modalità ... utilizza le funzioni di un gestore code IBM MQ per implementare JMS.

7 - Modalità normale con restrizioni

L'applicazione JMS utilizza la modalità normale con limitazioni. Questa modalità utilizza l'API JMS 2.0, ma non le nuove funzioni quali le sottoscrizioni condivise, il recapito ritardato o l'invio asincrono.

6 - modalità di migrazione

L'applicazione JMS utilizza la modalità di migrazione. In modalità migrazione, IBM MQ classes for JMS utilizza funzioni e algoritmi simili a quelli forniti con IBM WebSphere MQ 6.0.

non specificato (il valore predefinito)

Il client JMS utilizza un algoritmo per stabilire quale modalità di funzionamento viene utilizzata.

Il valore che si specifica per la proprietà **PROVIDERVERSION** deve essere una stringa. Se si specifica una delle opzioni 8, 7 o 6, è possibile farlo in uno qualsiasi dei seguenti formati:

- V.R.M.F
- V.R.M
- V.R
- V

dove V, R, M e F sono numeri interi maggiori o uguali a zero. Gli ulteriori valori R, M ed F sono facoltativi e sono disponibili per l'utilizzo qualora sia necessario un controllo più preciso. Ad esempio, se si desidera utilizzare un livello **PROVIDERVERSION** di 7, è possibile impostare **PROVIDERVERSION** = 7, 7.0, 7.0.0 o 7.0.0.0.

Tipi di oggetto factory di connessione

È possibile impostare la proprietà **PROVIDERVERSION** per i seguenti tipi di oggetto factory di connessione:

- MQConnectionFactory
- Factory di MQQueueConnection
- Factory MQTopicConnection
- MQXAConnectionFactory
- Factory MQXAQueueConnection
- Factory MQXAQueueConnection

- Factory MQXAQueueConnection
- Factory MQXATopicConnection

Per ulteriori informazioni su questi diversi tipi di factory di connessione, consultare [“Configurazione degli oggetti JMS e Jakarta Messaging utilizzando gli strumenti di amministrazione”](#) a pagina 711.

Concetti correlati

[Provider di messaggistica IBM MQ](#)

PROVIDERVERSION Modalità normale

La modalità normale utilizza tutte le funzioni di un gestore code IBM MQ per implementare JMS. Questa modalità è ottimizzata per utilizzare la API e la funzionalità JMS 2.0.

Il seguente diagramma di flusso mostra i controlli eseguiti dal client JMS per stabilire se è possibile creare una connessione in modalità normale.

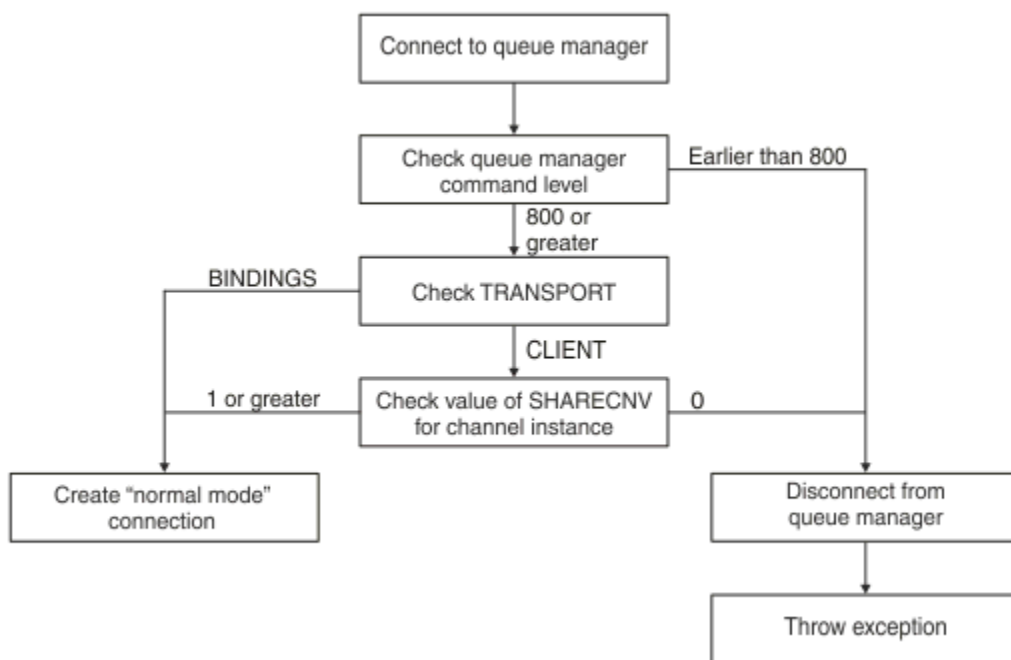


Figura 90. Modalità normale PROVIDERVERSION

Se il gestore code specificato nelle impostazioni del factory di connessione ha un livello di comando pari o superiore a 800 e la proprietà **TRANSPORT** del factory di connessione è impostata su BINDINGS, viene creata una connessione in modalità normale senza controllare ulteriori proprietà.

Se il gestore code specificato nelle impostazioni del factory di connessione ha un livello di comando di 800 o superiore e la proprietà **TRANSPORT** è impostata su CLIENT, viene controllata anche la proprietà **SHARECNV** sul canale di connessione del server. Questo controllo è necessario perché la modalità normale del provider di messaggistica di IBM MQ utilizza la funzione di condivisione delle conversazioni. Pertanto, affinché un tentativo di connessione in modalità normale abbia esito positivo, la proprietà **SHARECNV**, che controlla il numero di conversazioni che possono essere condivise, deve avere un valore pari o superiore a 1.

Se tutti i controlli mostrati nel diagramma di flusso hanno esito positivo, viene creata una connessione in modalità normale al gestore code e tutte le funzioni e l'API JMS 2.0, ovvero l'invio asincrono, la consegna ritardata e la sottoscrizione condivisa, possono essere utilizzate.

Un tentativo di creare una connessione in modalità normale non è riuscito per uno dei seguenti motivi:

- Il gestore code specificato nelle impostazioni della factory di connessione ha un livello di comando precedente a 800. In questo caso, il metodo `createConnection` non riesce con un'eccezione `JMSFMQ0003`.
- La proprietà **SHARECNV** sul canale di connessione server è impostata su 0. Se questa proprietà non ha un valore di 1 o superiore, il metodo `createConnection` non riesce con un'eccezione `JMSCC5007`.

Riferimenti correlati

Dipendenze tra proprietà di oggetti IBM MQ classes for JMS

[DEFINE CHANNEL \(SHARECNV proprietà\)](#)

[TRANSPORT](#)

PROVIDERVERSION modalità normale con limitazioni

La modalità normale con limitazioni utilizza l'API JMS 2.0 , ma non le nuove funzioni IBM MQ 8.0 o successive come le sottoscrizioni condivise, la consegna ritardata o l'invio asincrono.

Il seguente diagramma di flusso mostra i controlli eseguiti dal client JMS per determinare se è possibile creare una modalità normale con restrizioni di connessione.

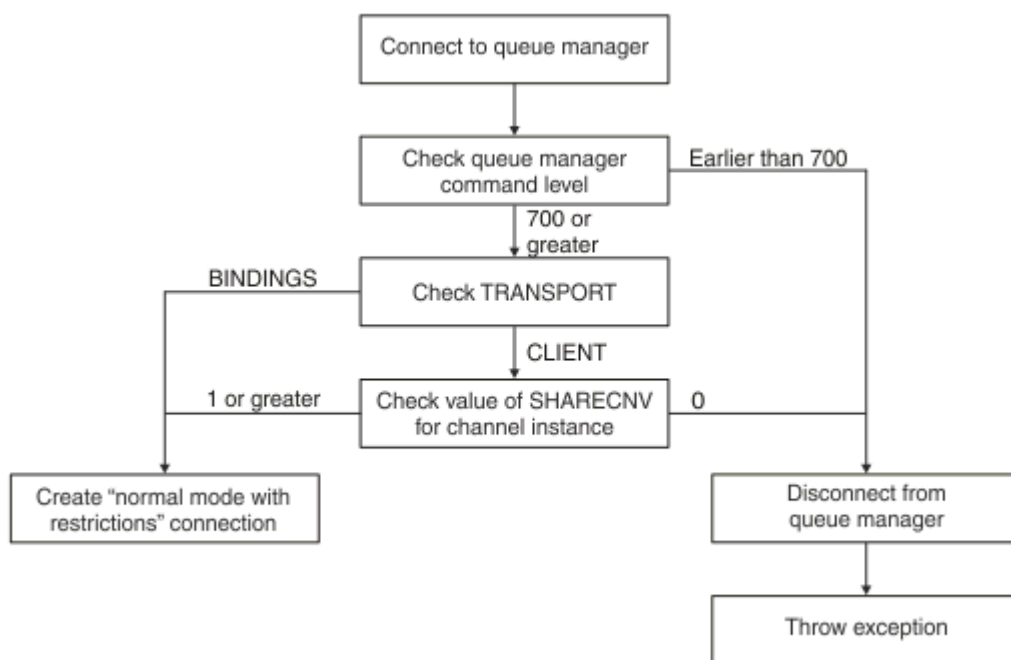


Figura 91. Modalità normale PROVIDERVERSION con restrizioni

Se il gestore code specificato nelle impostazioni della factory di connessione ha un livello di comando pari o superiore a 700 e la proprietà **TRANSPORT** della factory di connessione è impostata su **BINDINGS**, viene creata una connessione in modalità normale senza controllare ulteriori proprietà.

Se il gestore code specificato nelle impostazioni della factory di connessione ha un livello di comando di 700 o superiore e la proprietà **TRANSPORT** è impostata su **CLIENT**, viene controllata anche la proprietà **SHARECNV** sul canale di connessione del server. Questa verifica è necessaria perché la modalità normale del provider di messaggistica IBM MQ con limitazioni utilizza la funzionalità di condivisione delle conversazioni. Pertanto, per un normale tentativo di connessione in modalità normale con limitazioni, la proprietà **SHARECNV** , che controlla il numero di conversazioni che possono essere condivise, deve avere un valore pari o superiore a 1.

Se tutti i controlli mostrati nel diagramma di flusso hanno esito positivo, viene creata una modalità normale con restrizioni per la connessione al gestore code ed è quindi possibile utilizzare l'API JMS 2.0 , ma non le funzioni di invio asincrono, di consegna ritardata o di sottoscrizione condivisa.

Un tentativo di creare una modalità normale con restrizioni di connessione non riesce per uno dei seguenti motivi:

- Il gestore code specificato nelle impostazioni della factory di connessione ha un livello di comando precedente a 700. In questo caso, il metodo `createConnection` ha esito negativo con eccezione `JMSFCC5008`.
- La proprietà **SHARECNV** sul canale di connessione server è impostata su 0. Se questa proprietà non ha un valore di 1 o superiore, il metodo `createConnection` non riesce con un'eccezione `JMSCC5007`.

Riferimenti correlati

[Dipendenze tra proprietà di oggetti IBM MQ classes for JMS](#)

[DEFINE CHANNEL \(SHARECNV proprietà\)](#)

[TRANSPORT](#)

Modalità di migrazione **PROVIDERVERSION**

Per la modalità di migrazione, IBM MQ classes for JMS utilizza funzioni e algoritmi simili a quelli forniti con IBM WebSphere MQ 6.0, come la pubblicazione / sottoscrizione accodata, la selezione implementata sul lato client, i canali non multiplex e il polling utilizzato per implementare i listener.

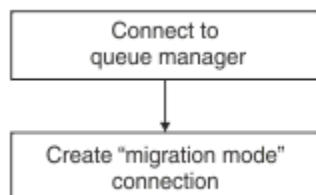


Figura 92. Modalità di migrazione **PROVIDERVERSION**

Se si desidera connettersi a WebSphere Message Broker 6.0 o WebSphere Message Broker 6.1 utilizzando IBM MQ Enterprise Transport versione 6.0, è necessario utilizzare la modalità di migrazione.

È possibile connettersi a un gestore code IBM MQ 8.0 utilizzando la modalità di migrazione, ma nessuna delle nuove funzionalità di un gestore code IBM MQ classes for JMS viene utilizzata, ad esempio la lettura anticipata o lo streaming. Se un client IBM MQ 8.0 o successivo si connette a un gestore code IBM MQ 8.0 o successivo su una piattaforma distribuita, **z/OS** o un gestore code IBM MQ for z/OS 8.0 o successivo, la selezione del messaggio viene effettuata dal gestore code anziché sul sistema client.

Se viene specificata la modalità di migrazione del fornitore di messaggistica IBM MQ e IBM MQ classes for JMS tenta di utilizzare una delle API JMS 2.0, la chiamata del metodo API ha esito negativo con l'eccezione `JMSCC5007`.

Riferimenti correlati

[Dipendenze tra proprietà di oggetti IBM MQ classes for JMS](#)

[TRANSPORT](#)

PROVIDERVERSION non specificato

Quando la proprietà **PROVIDERVERSION** di una factory di connessione non è specificata, il client JMS utilizza un algoritmo per determinare quale modalità operativa viene utilizzata per la connessione al gestore code. Una factory di connessione creata nello spazio nomi JNDI con una versione precedente di IBM MQ classes for JMS assume il valore non specificato quando la factory di connessione viene utilizzata con la nuova versione di IBM MQ classes for JMS.

Se la proprietà **PROVIDERVERSION** non è specificata, l'algoritmo viene utilizzato quando viene richiamato il metodo `createConnection`. L'algoritmo controlla un certo numero di proprietà della factory di connessione per stabilire se è richiesta la modalità normale del provider di messaggistica IBM MQ, la modalità normale con limitazioni o la modalità di migrazione del provider di messaggistica IBM MQ. La modalità normale viene sempre tentata prima e quindi la modalità normale con restrizioni. Se non è

possibile effettuare nessuno di questi tipi di connessione, il client JMS si disconnette dal gestore code e si connette nuovamente con il gestore code per tentare una connessione in modalità di migrazione.

Controllo delle proprietà **BROKERVER**, **BROKERQMGR**, **PSMODE** e **BROKERCONQ**

La verifica dei valori delle proprietà inizia con la proprietà **BROKERVER** come mostrato nella [Figura 1](#).

Se la proprietà **BROKERVER** è impostata su V1, la proprietà **TRANSPORT** viene selezionata successivamente come mostrato nella [Figura 2](#). Tuttavia, se la proprietà **BROKERVER** è impostata su V2, il controllo aggiuntivo mostrato nella [Figura 1](#) viene eseguito prima del controllo della proprietà **TRANSPORT**.

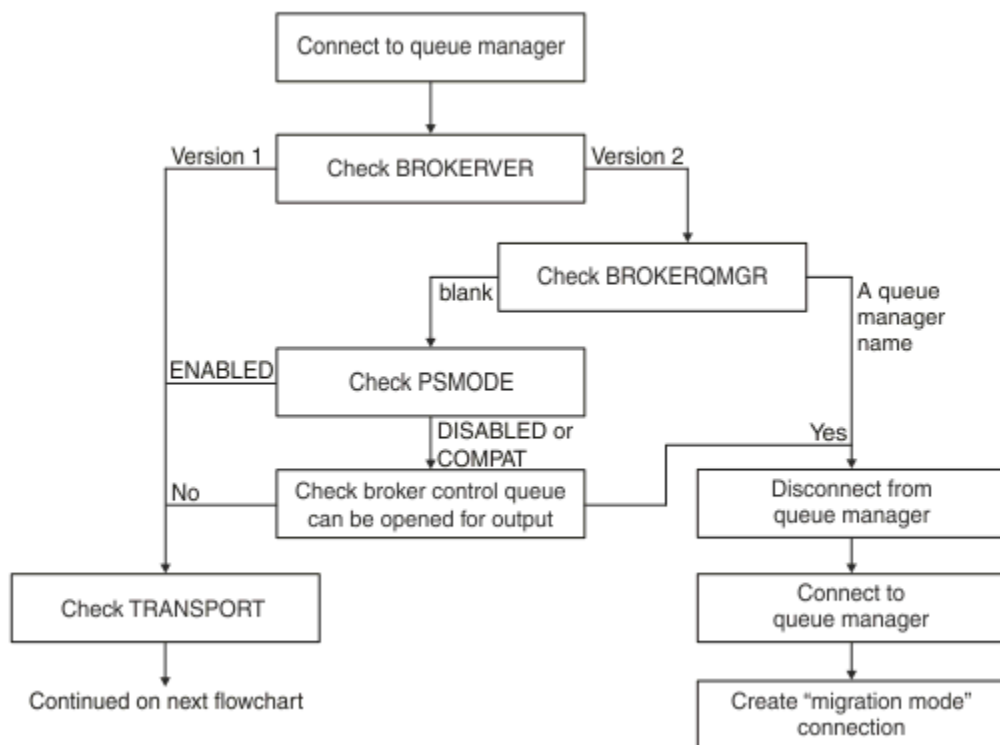


Figura 93. PROVIDERVERSION non specificato

Se la proprietà **BROKERVER** è impostata su V2, per rendere possibile una connessione in modalità normale, la proprietà **BROKERQMGR** deve essere vuota. Inoltre, l'attributo **PSMODE** sul gestore code deve essere impostato su **ENABLED** oppure la coda di controllo del broker specificata dalla proprietà **BROKERCONQ** non deve essere aperta per l'output.

Se i valori delle proprietà sono impostati come richiesto per una connessione in modalità normale, il controllo passa alla proprietà **TRANSPORT** come mostrato nella [Figura 2](#).

Se i valori delle proprietà non sono impostati come richiesto per una connessione in modalità normale, il client JMS si disconnette dal gestore code, quindi si riconnette e crea una connessione in modalità di migrazione. Ciò si verifica nei seguenti casi:

- Se la proprietà **BROKERQMGR** è vuota e l'attributo **PSMODE** sul gestore code è impostato su **COMPAT** o **DISABLED** e la coda di controllo broker specificata dalla proprietà **BROKERCONQ** può essere aperta per l'output (ossia, **MQOPEN** per l'output riuscito).
- Se la proprietà **BROKERQMGR** specifica un nome coda.

Controllo della proprietà **TRANSPORT** e del livello di comando

La [Figura 2](#) mostra i controlli effettuati per il livello di comando e la proprietà **TRANSPORT** del gestore code.

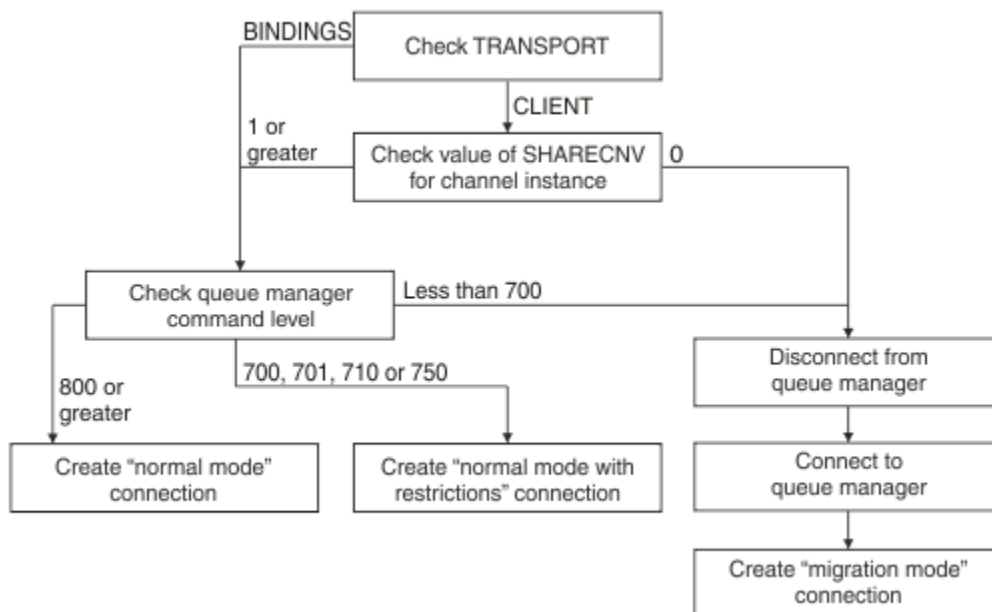


Figura 94. PROVIDERVERSION non specificato (continua)

Una connessione in modalità normale viene creata in uno dei seguenti casi:

- La proprietà **TRANSPORT** della factory di connessione è impostata su BINDINGS e il gestore code ha un livello di comando di 800 o superiore.
- La proprietà **TRANSPORT** è impostata su CLIENT, la proprietà **SHARECNV** sul canale di connessione server ha un valore pari o superiore a 1 e il gestore code ha un livello di comando pari o superiore a 800.

Se il gestore code ha un livello di comando di 750, viene creata una modalità normale con limitazioni di connessione al gestore code.

Una connessione in modalità di migrazione viene creata anche se la proprietà **TRANSPORT** è impostata su CLIENT e la proprietà **SHARECNV** sul canale di connessione del server ha il valore 0.

Riferimenti correlati

Dipendenze tra proprietà di oggetti IBM MQ classes for JMS

[ALTER QMGR \(attributo PSMODE\)](#)

[BROKERCONQ](#)

[BROKERQMGR](#)

[BROKERVER](#)

[DEFINE CHANNEL \(SHARECNV proprietà\)](#)

[TRANSPORT](#)

Configurazione delle informazioni sulla versione del provider in WebSphere Application Server

Per configurare le informazioni sulla versione del fornitore in WebSphere Application Server, è possibile utilizzare la console di gestione o i comandi wsadmin.

Procedura

Per configurare le informazioni sulla versione del provider per un factory di connessione IBM MQ o per un oggetto della specifica di attivazione in WebSphere Application Server, consultare *Informazioni correlate* per i link a ulteriori informazioni nella documentazione del prodotto WebSphere Application Server .

Informazioni correlate per WebSphere Application Server 8.5.5

[Impostazioni della factory di connessione del fornitore di messaggistica IBM MQ](#)

[**createWMQConnectionFactory** comando](#)

[Impostazioni della specifica di attivazione del fornitore di messaggistica IBM MQ](#)

[**createWMQActivationSpec** comando](#)

Informazioni correlate per WebSphere Application Server 8.0.0

[Impostazioni della factory di connessione del fornitore di messaggistica IBM MQ](#)

[**createWMQConnectionFactory** comando](#)

[Impostazioni della specifica di attivazione IBM MQ](#)

[**createWMQActivationSpec** comando](#)

Informazioni correlate per WebSphere Application Server 7.0.0

[Impostazioni della factory di connessione del fornitore di messaggistica IBM MQ](#)

[**createWMQConnectionFactory** comando](#)

[Impostazioni della specifica di attivazione IBM MQ](#)

[**createWMQActivationSpec** comando](#)

Rimozione di sottoscrizioni durevoli WebSphere Application Server

Quando si utilizza il provider di messaggistica IBM MQ con WebSphere Application Server 7.0 e WebSphere Application Server 8.0, le sottoscrizioni durevoli create dalle applicazioni MDB (message - driven bean) associate alle specifiche di attivazione non vengono rimosse. Le sottoscrizioni durature possono essere rimosse utilizzando il programma di utilità della riga comandi IBM MQ Explorer o IBM MQ .

Informazioni su questa attività

Un'applicazione MDB (message - driven bean) che rimuove una sottoscrizione durevole può essere configurata per utilizzare una porta listener o una specifica di attivazione, purché l'applicazione sia in esecuzione all'interno di un'istanza WebSphere Application Server 7.0 o WebSphere Application Server 8.0 che utilizza la modalità normale del provider di messaggistica [IBM MQ](#) per connettersi a IBM MQ.

Se l'applicazione MDB (message - driven bean) è collegata a una porta listener, il provider di messaggistica IBM MQ crea la sottoscrizione durevole per l'applicazione la prima volta che l'applicazione viene avviata. La sottoscrizione durevole viene rimossa quando l'applicazione bean basato sui messaggi viene disinstallata da un server delle applicazioni e il server delle applicazioni viene riavviato.

Un'applicazione MDB (message - driven bean) collegata a una specifica di attivazione funziona in un modo leggermente diverso. La sottoscrizione durevole viene creata per l'applicazione la prima volta che l'applicazione viene avviata. Tuttavia, la sottoscrizione durevole non viene rimossa quando l'applicazione viene disinstallata e il server delle applicazioni riavviato.

Ciò può portare a un numero di sottoscrizioni durevoli che rimangono su un motore di pubblicazione / sottoscrizione IBM MQ per le applicazioni che non sono più installate in un sistema WebSphere Application Server . Queste sottoscrizioni sono note come "sottoscrizioni orfane" e possono causare problemi sul gestore code quando il motore di pubblicazione / sottoscrizione è in esecuzione.

Quando un messaggio viene pubblicato su un argomento, il motore di pubblicazione / sottoscrizione IBM MQ crea una copia di tale messaggio per ciascuna sottoscrizione durevole registrata su tale argomento e lo inserisce in una coda interna. Le applicazioni che utilizzano tale sottoscrizione duratura raccoglieranno e consumeranno il messaggio da questa coda interna.

Se l'applicazione MDB (message - driven bean) che stava utilizzando tale sottoscrizione durevole non è più installata, le copie dei messaggi pubblicati per l'applicazione continueranno ad essere eseguite. Tuttavia, questi messaggi non saranno mai elaborati, il che significa che potrebbe esserci un numero elevato di messaggi rimanenti nella coda interna che non verranno mai rimossi.

Prima di iniziare

Le sottoscrizioni registrate con il motore di pubblicazione / sottoscrizione IBM MQ avranno un nome sottoscrizione associato.

Le sottoscrizioni durevoli create dal provider di messaggistica WebSphere Application Server IBM MQ per i bean basati sui messaggi collegati alle specifiche di attivazione avranno un nome sottoscrizione nel seguente formato:

```
JMS:queue manager name:client identifier:subscription name
```

dove:

nome gestore code

Questo è il nome del gestore code IBM MQ in cui è in esecuzione il motore di pubblicazione / sottoscrizione.

Identificativo client

Questo è il valore della proprietà ID client della specifica di attivazione a cui è collegato il bean basato sui messaggi.

Nome sottoscrizione

Questo è il valore della proprietà della specifica di attivazione Nome sottoscrizione per la specifica di attivazione che l'applicazione MDB (message - driven bean) è stata configurata per utilizzare.

Ad esempio, si supponga di disporre di una specifica di attivazione impostata per la connessione al gestore code testQM. La specifica di attivazione ha le seguenti proprietà impostate:

- ID client = testClientID
- Nome sottoscrizione = durableSubscription1

Se un bean basato sui messaggi che esegue una sottoscrizione durevole è collegato a questa specifica di attivazione, viene creata una sottoscrizione sul IBM MQ motore di pubblicazione / sottoscrizione sul gestore code testQM che ha il seguente nome sottoscrizione:

- JMS:testQM:testClientID:durableSubscription1

Le sottoscrizioni che sono state registrate con il motore di pubblicazione / sottoscrizione IBM MQ per un determinato gestore code possono essere visualizzate in uno dei due seguenti modi:

- La prima opzione consiste nell'utilizzare MQ Explorer. Quando MQ Explorer è stato connesso ad un gestore code utilizzato per il lavoro di pubblicazione / sottoscrizione, l'elenco dei sottoscrittori attualmente registrati con il motore di pubblicazione / sottoscrizione può essere visualizzato facendo clic sulla voce IBM WebSphere MQ ->queue manager name-> Subscriptions nel riquadro di navigazione.
- L'altro modo per visualizzare le sottoscrizioni che sono state registrate con un motore di pubblicazione / sottoscrizione è utilizzare il IBM MQ programma di utilità della riga comandi **runmqsc** ed eseguire il comando **display sub**. A tale scopo, visualizzare un prompt dei comandi, passare alla directory *WebSphere MQ\bin* e immettere il comando seguente per avviare **runmqsc**:

```
- runmqsc queue manager name
```

Una volta avviato il programma di utilità **runmqsc**, immettere il seguente comando per elencare tutte le sottoscrizioni durevoli attualmente registrate con il motore di pubblicazione / sottoscrizione in esecuzione sul gestore code a cui è connesso **runmqsc**:

```
- display sub(*) durable
```

Per verificare se le sottoscrizioni durevoli registrate con i motori di pubblicazione / sottoscrizione sono ancora attive:

1. Generare l'elenco delle sottoscrizioni durevoli che sono state registrate con il motore di pubblicazione / sottoscrizione.
2. Per ogni sottoscrizione durevole:

- Controllare il nome della sottoscrizione per il sottoscrittore durevole e annotare il valore *identificativo client* e *nome sottoscrizione*.
- Esaminare i sistemi WebSphere Application Server che si stanno collegando a questo motore di pubblicazione / sottoscrizione. Verificare se sono state definite specifiche di attivazione che hanno la proprietà ID client che corrisponde al valore *identificativo client* e la proprietà del nome sottoscrizione che corrisponde al *nome sottoscrizione*.
- Se non viene trovata alcuna specifica di attivazione che abbia le proprietà ID client e nome sottoscrizione che corrispondono ai campi *identificativo client* e *nome sottoscrizione* nel nome sottoscrizione IBM MQ, non ci sono specifiche di attivazione che utilizzano questa sottoscrizione durevole. La sottoscrizione durevole può essere eliminata.
- Se è stata definita una specifica di attivazione che corrisponde al nome della sottoscrizione durevole, il controllo finale che deve essere effettuato è quello di verificare se esiste un'applicazione MDB (message - driven bean) che utilizza questa specifica di attivazione. Per far ciò:
 - Prendere nota del nome JNDI per la specifica di attivazione che ha preso la sottoscrizione durevole che si sta attualmente cercando.
 - Visualizzare il riquadro Configurazione nella console di gestione WebSphere Application Server per ogni applicazione bean basata sui messaggi installata.
 - Fare clic sul collegamento Bind del listener MDB (Message Driven Bean) nel riquadro Configurazione.
 - Viene visualizzata una tabella con le informazioni sull'applicazione MDB (message - driven bean). Se il pulsante di opzione della specifica di attivazione è selezionato nella colonna Bind e il campo del nome della risorsa di destinazione JNDI contiene il nome JNDI per la specifica di attivazione che ha eliminato la sottoscrizione durevole, la sottoscrizione è ancora in uso e non può essere eliminata.
 - Se non è possibile trovare alcuna applicazione bean basata sui messaggi che utilizza la specifica di attivazione, è possibile eliminare la sottoscrizione durevole.

Procedura

Una volta identificata una sottoscrizione duratura "orfana", è possibile eliminarla utilizzando il IBM MQ Explorer o il IBM MQ programma di utilità della riga comandi **runmqsc**.

Per eliminare una sottoscrizione durevole "orfana" utilizzando il IBM MQ Explorer:

1. Evidenziare la voce per la sottoscrizione
2. Fare clic con il tasto destro del mouse sulla voce e selezionare **Elimina ...** dal menu. Viene visualizzata una finestra di conferma.
3. Verificare che il nome della sottoscrizione visualizzato nella finestra di conferma sia corretto e fare clic su **Sì**.

Il IBM MQ Explorer ora elimina la sottoscrizione dal motore di pubblicazione / sottoscrizione e ripulisce tutte le risorse interne associate ad essa (ad esempio i messaggi non elaborati che sono stati pubblicati per l'argomento su cui è stata registrata la sottoscrizione durevole).

Per eliminare una sottoscrizione durevole "orfana" utilizzando l'utilità della riga comandi IBM MQ **runmqsc**, è necessario eseguire il comando **delete sub** :

1. Aprire una sessione del prompt dei comandi
2. Passare alla directory *IBM MQ\bin*
3. Immettere il seguente comando per avviare **runmqsc**:

```
runmqsc queue manager name
```

4. Una volta avviato il programma di utilità **runmqsc** , immettere:

```
delete sub(Subscription name)
```

dove *Nome sottoscrizione* è il nome della sottoscrizione durevole, che assume il seguente formato:

- `JMS:queue manager name:client identifier:subscription name`

Configurazione di Managed File Transfer

È possibile configurare le funzioni di Managed File Transfer dopo l'installazione.

Puoi sfruttare le soluzioni di alta disponibilità IBM MQ per migliorare la resilienza della tua configurazione Managed File Transfer . Se gli agent utilizzano gestori code di dati replicati (RDQM), è necessario configurarli per utilizzare la funzione di indirizzo IP mobile. Ciò significa che gli agenti utilizzano lo stesso indirizzo IP per comunicare con una delle tre istanze RDQM attualmente in esecuzione e si riconnettono automaticamente in caso di failover (consultare [Alta disponibilità RDQM](#) e [Creazione ed eliminazione di un indirizzo IP mobile](#)). Se si utilizza la soluzione del gestore code a più istanze, le applicazioni utilizzano un diverso indirizzo IP per comunicare con ciascuna istanza, che viene gestita dalla riconnessione client in caso di failover (consultare [“Gestori code a più istanze”](#) a pagina 524 e [“Riconnessione canale e client”](#) a pagina 580).

Concetti correlati

[Suggerimenti e suggerimenti per l'utilizzo di Managed File Transfer](#)

Attività correlate

[Monitoraggio delle risorse MFT](#)

[Personalizzare MFT con uscite utente](#)

[Configurazione di MQMFTCredentials.xml](#)

[protezioneManaged File Transfer](#)

[Specifica dei programmi da eseguire con MFT](#)

[Risoluzione dei problemiManaged File Transfer](#)

[AmministrazioneManaged File Transfer](#)

Riferimenti correlati

[MFT Comandi](#)

[Il file MFTagent.properties](#)

[Ripristino e riavvio di MFT](#)

Opzioni di configurazione MFT su Multiplatforms

Managed File Transfer fornisce una serie di file delle proprietà che contengono le informazioni chiave sull'impostazione e sono richiesti per l'operazione. Questi file delle proprietà si trovano nella directory di configurazione definita quando è stato installato il prodotto.

È possibile avere più serie di opzioni di configurazione, ogni serie di opzioni di configurazione contiene una serie di directory e file delle proprietà. I valori definiti in questi file delle proprietà ... vengono utilizzati come parametri predefiniti per tutti i comandi Managed File Transfer , a meno che non si specifichi esplicitamente un altro valore sulla riga comandi.

Per modificare la serie predefinita di opzioni di configurazione che si stanno utilizzando, è possibile utilizzare il comando **ftChangeDefaultConfigurationOptions** . Per modificare la serie di opzioni di configurazione che si sta utilizzando per un singolo comando, è possibile utilizzare il parametro **-p** con qualsiasi comando Managed File Transfer .

Il nome di una serie di opzioni di configurazione è il nome del gestore code di coordinamento e si consiglia di non modificarlo. Tuttavia, è possibile modificare il nome di una serie di opzioni di configurazione, ma è necessario modificare il nome delle directory `config` e `logs` . Nei seguenti esempi, il nome della serie di opzioni di configurazione è rappresentato come `coordination_qmgr_name`.

Struttura di directory delle opzioni di configurazione

Quando si configura il prodotto, le directory e i file delle proprietà vengono creati nella seguente struttura nella directory di configurazione. È inoltre possibile modificare tali directory e file delle proprietà con i seguenti comandi: **fteSetupCoordination**, **fteSetupCommands**, **fteChangeDefaultConfiguration** e **fteCreateAgent**.

```
MQ_DATA_PATH/mqft/  
  config/  
    coordination_qmgr_name/  
      coordination.properties  
      command.properties  
    agents/  
      agent_name/  
        agent.properties  
        exits  
      loggers/  
        logger_name  
        logger.properties  
  installations/  
    installation_name/  
      installation.properties
```

La directory *coordination_qmgr_name* è una directory di opzioni di configurazione. Nella directory di configurazione può essere presente più di una directory di opzioni di configurazione. La directory *agent_name* è una directory dell'agent. Oltre a contenere il file *agent.properties*, questa directory contiene la directory *exits*, che è il percorso predefinito per le routine di uscita utente e i diversi file XML generati dai comandi **fteCreateBridgeAgent** e **fteCreateCDAgent**. Può esistere più di una directory agent nella directory *agents* di una serie di opzioni di configurazione.

File Properties

installation.properties

Il file *installation.properties* specifica il nome della serie predefinita di opzioni di configurazione. Questa voce punta Managed File Transfer ad una serie strutturata di indirizzi e file delle proprietà che contengono la configurazione da utilizzare. Generalmente, il nome di una serie di opzioni di configurazione è il nome del gestore code di coordinamento associato. Per ulteriori informazioni sul file *installation.properties*, consultare [Il file MFT installation.properties](#).

coordination.properties

Il file *coordination.properties* specifica i dettagli di collegamento al gestore code di coordinamento. Poiché diverse installazioni di Managed File Transfer potrebbero condividere lo stesso gestore code di coordinamento, è possibile utilizzare un link simbolico a un file *coordination.properties* comune su un'unità condivisa. Per ulteriori informazioni sul file *coordination.properties*, consultare [Il file MFT coordination.properties](#).

command.properties

Il file MFT *command.properties* specifica il gestore code comandi a cui connettersi quando si immettono i comandi e le informazioni richieste da Managed File Transfer per contattare tale gestore code. Per ulteriori informazioni sul file *command.properties*, consultare [Il file MFT command.properties](#).

agent.properties

Ogni Managed File Transfer Agent ha il proprio file delle proprietà, *agent.properties*, che deve contenere le informazioni che un agente utilizza per connettersi al proprio gestore code. Il file *agent.properties* può anche contenere proprietà che modificano il comportamento dell'agent. Per ulteriori informazioni relative al file *agent.properties*, consultare [Il file MFT agent.properties](#).

logger.properties

Il file *logger.properties* specifica le proprietà di configurazione per i logger. Per ulteriori informazioni sul file *logger.properties*, consultare [Proprietà di configurazione del programma di registrazioneMFT](#).

File delle proprietà e codepage

Il contenuto di tutti i file delle proprietà Managed File Transfer deve rimanere in inglese (Stati Uniti) a causa di una limitazione di Java. Se si modificano i file delle proprietà su un sistema non inglese (Stati Uniti), è necessario utilizzare le sequenze di escape Unicode.

Riferimenti correlati

[Proprietà SSL/TLS per MFT](#)

[Java proprietà di sistema per MFT](#)

[Opzioni fteChangeDefaultConfiguration](#)

[Comandi fteSetup: creare il file MFT command.properties](#)

[Coordinamento fteSetup](#)

[Agent fteCreate](#)

MFT configuration options on z/OS

The Managed File Transfer configuration options on z/OS are the same as the options for distributed platforms.

For more information about configuration options on [Multiplatforme](#), see “[Opzioni di configurazione MFT su Multiplatforms](#)” on page 744.

On z/OS, the configuration location is defined by the environment variable BFG_DATA. If a configuration does not already exist under the z/OS UNIX System Services directory that is referenced by BFG_DATA, the BFGCUSTOM JCL script of an MFT command PDSE library data set generates the jobs required to create the configuration. The configuration is then created when you run these generated jobs. Configuration creation relies on BFG_DATA referencing an existing directory that is accessible.

You can also create and maintain a configuration by using the same **fte** commands that are available on both Multiplatforms and z/OS. For a list of the **fte** commands, see [MFT commands](#).

Related concepts

[“Opzioni di configurazione MFT su Multiplatforms” on page 744](#)

Managed File Transfer fornisce una serie di file delle proprietà che contengono le informazioni chiave sull'impostazione e sono richiesti per l'operazione. Questi file delle proprietà si trovano nella directory di configurazione definita quando è stato installato il prodotto.

[“Creating an agent” on page 762](#)

You need to copy the PDSE to make the agent-specific PDSE, for example *user.MFT.AGENT1*. Copy the PDSE from a previous agent or logger configuration, if they exist. If this is your first configuration, copy the PDSE supplied with MFT.

[“Defining the coordination queue manager” on page 761](#)

Managed File Transfer requires a queue manager to be created that acts as the coordination queue manager.

Related tasks

 [Configuring MQMFTCredentials.xml on z/OS](#)

[“Updating an existing MFT Agent or Logger command data set on z/OS” on page 764](#)

You can update an Managed File Transfer command PDSE library data set that is created from the Managed File Transfer command template data set.

Download e configurazione di Redistributable Managed File Transfer components

Il Redistributable Managed File Transfer package fornisce il Redistributable Managed File Transfer Agent, che puoi configurare per connetterti ad un'infrastruttura IBM MQ esistente e consentire agli utenti di trasferire i file senza dover installare IBM MQ. Da IBM MQ 9.3.0, il package ridistribuibile include anche Redistributable Managed File Transfer Logger.

Prima di iniziare

Per informazioni sui termini di licenza redistribuibili per Redistributable Managed File Transfer Agent e Redistributable Managed File Transfer Logger, consultare [IBM MQ Componenti redistribuibili](#).

I componenti Redistributable Managed File Transfer package forniscono la funzionalità di Managed File Transfer con le seguenti eccezioni:

- Per Redistributable Managed File Transfer Agent, la connessione in modalità bind ai gestori code di coordinazione, comando e agent non è supportata, è necessario utilizzare la connessione in modalità client. Quando si immettono comandi, è necessario fornire i parametri facoltativi quando si utilizza il Managed File Transfer installato come parte di IBM MQ: host del gestore code, porta, nome e nome del canale.
- Redistributable Managed File Transfer Logger supporta solo i logger di tipo FILE, che si collegano in modalità client solo al gestore code di coordinamento. La connessione in modalità client al gestore code di coordinamento per un programma di registrazione database non è supportata. Se si richiede una connessione in modalità bind, è necessario utilizzare un'installazione standard di IBM MQ.
- Da IBM MQ 9.3.0, il comando **fteCreateCDAgent.cmd** non è incluso. Per un elenco completo dei comandi disponibili, consultare [Serie comandi MFT installati](#).
- Managed File Transfer Connect:Direct non è supportato.
- IBM MQ Explorer non è incluso.

Windows È necessario installare le librerie Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019, disponibili da Microsoft, sul sistema per utilizzare Redistributable Managed File Transfer Agent. Consultare [Gli ultimi download Visual C++ supportati](#).

Da IBM MQ 9.3.0, le librerie Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 sono richieste anche per Redistributable Managed File Transfer Logger.

Nota: Advanced Message Security non è supportato con Redistributable Managed File Transfer package.

Informazioni su questa attività

Facoltativamente, puoi scaricare Redistributable Managed File Transfer package e configurare Redistributable Managed File Transfer Agent per la connessione a un'infrastruttura IBM MQ esistente per consentire agli utenti di trasferire i file tra il proprio ambiente locale e l'infrastruttura IBM MQ esistente senza dover installare IBM MQ per ottenere la funzione Managed File Transfer.

Da IBM MQ 9.3.0, Redistributable Managed File Transfer package include anche Redistributable Managed File Transfer Logger, che consente di impostare un programma di registrazione file per connettersi in modalità client al gestore code di coordinamento.

Procedura

1. Scarica il [Pacchetto agent IBM MQ redistribuibile Managed File Transfer](#) da Fix Central.

a) Scegliere il pacchetto per il proprio sistema operativo.

I nomi dei file di archivio o .zip descrivono il contenuto del file e i livelli di manutenzione equivalenti. I nomi file sono nel seguente formato:

- **Windows** V.R.M.F-IBM-MQFA-Redist-Win64
- **Linux** V.R.M.F-IBM-MQFA-Redist-LinuxX64
- **Linux** V.R.M.F-IBM-MQFA-Redist-LinuxS390X
- **Linux** V.R.M.F-IBM-MQFA-Redist-LinuxPPC64LE

dove V.R.M.F è il numero di versione, ad esempio 9.2.0.0 o 9.2.1.0.

b) Identificare la directory in cui si desidera estrarre il pacchetto, ad esempio:

- **Windows** C:\MFTZ
- **Linux** /home/MFTZ

2. Estrarre il contenuto del package scaricato:

- **Windows** Su Windows, utilizzare gli strumenti di Windows Explorer per eseguire l'estrazione.
- **Linux** Su Linux, estrarre e decomprimere come segue:

```
gunzip V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar.gz
```

e poi

```
tar xvf V.R.M.F-IBM-MQFA-Redist-LinuxX64.tar
```

dove *V.R.M.F* è il numero di versione, ad esempio 9.3.0.0 o 9.3.1.0.

Vengono create le seguenti directory:

- **Linux** **Windows** bin: contiene tutti i comandi MFT richiesti
- **Windows** bin64: contiene le librerie richieste necessarie per il supporto SO Windows a 64 bit
- **Linux** **Windows** java: contiene le librerie IBM JRE e IBM MQ .
- **Linux** **Windows** licenses: contiene i file di licenza
- **Windows** META-INF: contiene i file che hanno informazioni sulla firma del codice
- **Linux** **Windows** mqft: contiene le directory ant e lib richieste per il supporto Ant e per il supporto della funzione MFT principale
- **Linux** **Windows** swtag: contiene il file swidtag richiesto dai gestori delle licenze per identificare le installazioni sulla macchina

Operazioni successive

È possibile configurare un Managed File Transfer Agent. Per i passi successivi, consultare [“Creazione della configurazione iniziale per Redistributable Managed File Transfer Agent”](#) a pagina 748.

Da IBM MQ 9.3.0, è anche possibile configurare un Managed File Transfer Logger. Per i passi successivi per configurare il programma di registrazione, consultare [“Creazione della configurazione iniziale per Redistributable Managed File Transfer Logger”](#) a pagina 751.

Riferimenti correlati

[Possibili errori durante la configurazione di Redistributable Managed File Transfer components](#)

Linux **Windows** Creazione della configurazione iniziale per Redistributable Managed File Transfer Agent

È possibile configurare un Managed File Transfer Agent per connettersi a una configurazione IBM MQ esistente.

Prima di iniziare

Assicurarsi di scaricare ed estrarre il contenuto del pacchetto Redistributable Managed File Transfer Agent . Per ulteriori informazioni, consultare [“Download e configurazione di Redistributable Managed File Transfer components”](#) a pagina 746.

Informazioni su questa attività

Si crea innanzitutto l'ambiente di cui Redistributable Managed File Transfer Agent ha bisogno. È quindi possibile impostare la connettività con il gestore code in esecuzione sul server IBM MQ , quindi configurare un agent e il gestore code agent, prima di avviare e verificare l'agent.

Da IBM MQ 9.3.0, l'ambiente che crei è condiviso con Redistributable Managed File Transfer Logger. Per ulteriori informazioni, consultare [“Creazione della configurazione iniziale per Redistributable Managed File Transfer Logger”](#) a pagina 751.

Procedura

1. Creare l'ambiente per Redistributable Managed File Transfer Agent.

Quando si esegue il comando **fteCreateEnvironment**, viene creata la directory di dati MFT con le informazioni di configurazione per agent MFT . Accertarsi di trovarsi nella directory di bin creata quando è stato estratto il componente Redistributable Managed File Transfer Agent scaricato. Esegui il seguente comando:

- Windows

```
fteCreateEnvironment.cmd -d datapath location
```

- Linux

```
./fteCreateEnvironment -d datapath location
```

Questo comando utilizza i seguenti parametri facoltativi:

-d

Questo parametro specifica l'ubicazione per il percorso dati in cui viene creata, memorizzata e conservata la configurazione MFT . Se si esegue **fteCreateEnvironment** senza specificare l'ubicazione dei dati, la directory mftdata viene creata nell'ubicazione in cui viene estratto Redistributable Managed File Transfer Agent .

Nota: Se l'agent ridistribuibile verrà eseguito come un servizio Windows, la variabile di ambiente **BFG_DATA** deve essere impostata nell'ambiente di sistema perché il servizio funzioni.

-n nome installazione

Questo parametro viene utilizzato per specificare il nome di una installazione di IBM MQ o un nome univoco.

Esempi di situazioni in cui si potrebbe voler utilizzare questo parametro sono:

- Se si desidera testare rapidamente una nuova funzione o funzione utilizzando il pacchetto ridistribuibile con la configurazione esistente in cui gli agent sono stati configurati per connettersi al gestore code solo in modalità client. Tenere presente che questo parametro non si applica ad alcun agent configurato per la connessione a un gestore code in modalità bind.
- Se si sta eseguendo la migrazione da un'installazione standard di Managed File Transfer a un package Redistributable Managed File Transfer Agent e si desidera utilizzare la stessa configurazione di quella creata dall'installazione standard. Questo è il caso in cui Managed File Transfer standard è stato installato ma si sta connettendo a un gestore code dell'agent in esecuzione su un'altra macchina.

La variabile del nome di installazione predefinita è **BFG_INSTALLATION_NAME**.

Per ulteriori informazioni relative al comando **fteCreateEnvironment** , consultare [fteCreateEnvironment \(set up environment for Redistributable Managed File Transfer Agent\)](#).

È anche possibile impostare la variabile di ambiente **BFG_DATA** con l'ubicazione del percorso dati:

```
BFG_DATA=Datapath location
```

Prima di creare, avviare e arrestare un agent o qualsiasi altro comando, è necessario verificare che la variabile `BFG_DATA` sia impostata sul percorso dati corretto.

2. Impostare la connettività IBM MQ .

a) Impostare il gestore code di coordinamento utilizzando il comando **fteSetupCoordination** .

Il comando **fteSetupCoordination** crea il set attivo richiesto per i gestori code di coordinamento e le directory necessarie per un'ulteriore configurazione. Redistributable Managed File Transfer Agent funziona in modalità client, quindi è necessario fornire ulteriori parametri con questo comando per evitare un errore, poiché la modalità di bind non è supportata.

```
fteSetupCoordination -coordinationQMGr PRMFTDEM02
                    -coordinationQMGrHost 9.121.59.233 -coordinationQMGrPort 3002
                    -coordinationQMGrChannel SYSTEM.DEF.SVRCONN
```

Per ulteriori dettagli e passi per l'utilizzo del comando **fteSetupCoordination** , consultare [fteSetupCoordination](#). Per informazioni su come configurare il gestore code di coordinamento, consultare [“Configurazione del gestore code di coordinamento per MFT”](#) a pagina 790.

b) Creare e impostare il gestore code comandi:

```
fteSetupCommands -p PRMFTDEM02 -connectionQMGrHost 9.121.59.233
                 -connectionQMGrPort 3002 -connectionQMGrChannel SYSTEM.DEF.SVRCONN
                 -connectionQMGr PRMFTDEM02 -f
```

Per ulteriori dettagli e passi per l'utilizzo del comando **fteSetupCommands** , consultare [ComandifteSetup: creare il file MFT command.properties](#).

3. Creare una definizione di agent MFT per un endpoint.

```
fteCreateAgent -p PRMFTDEM02 -agentQMGrHost 9.121.59.233
               -agentQMGrPort 3002 -agentQMGrChannel SYSTEM.DEF.SVRCONN
               -agentName AGENT.TRI.BANK -agentQMGr PRMFTDEM02 -f
```

Per ulteriori informazioni sull'utilizzo del comando **fteCreateAgent** per configurare un agent e il gestore code dell'agent, consultare [fteCreateAgent](#).

Nota: È necessario utilizzare i comandi MQSC visualizzati come parte dell'emissione del comando per definire gli oggetti dell'agente sul gestore code dell'agente, altrimenti le istruzioni del passo [“4”](#) a pagina 750 non funzioneranno.

Nei passi [“2”](#) a pagina 750 e [“3”](#) a pagina 750 per ogni agent, si creano definizioni di argomenti e code sul gestore code dell'agent.

4. Avviare l'agent e si è pronti a trasferire i file.

```
fteStartAgent -p PRMFTDEM02 AGENT.TRI.BANK
```

È possibile verificare lo stato dell'agent eseguendo il seguente comando:

```
fteListAgents
```

Per ulteriori dettagli sull'utilizzo del comando **fteListAgents** , consultare [fteListAgents](#).

Operazioni successive

Se si desidera configurare Redistributable Managed File Transfer Logger, completare la procedura in [“Creazione della configurazione iniziale per Redistributable Managed File Transfer Logger”](#) a pagina 751.

Concetti correlati

[“Configurazione di Managed File Transfer”](#) a pagina 744

È possibile configurare le funzioni di Managed File Transfer dopo l'installazione.

[“Opzioni di configurazione MFT su Multiplatforms”](#) a pagina 744

Managed File Transfer fornisce una serie di file delle proprietà che contengono le informazioni chiave sull'impostazione e sono richiesti per l'operazione. Questi file delle proprietà si trovano nella directory di configurazione definita quando è stato installato il prodotto.

Riferimenti correlati

[fteCreateTransfer](#): avviare un nuovo trasferimento file

Creazione della configurazione iniziale per Redistributable Managed File Transfer Logger

È possibile configurare un tipo FILE Managed File Transfer Logger per connettersi a un gestore code di coordinamento in modalità client.

Prima di iniziare

Assicurarsi di scaricare ed estrarre il contenuto del pacchetto Redistributable Managed File Transfer Agent . Da IBM MQ 9.3.0, questo pacchetto include anche Redistributable Managed File Transfer Logger. Per ulteriori informazioni, fare riferimento a [“Download e configurazione di Redistributable Managed File Transfer components”](#) a pagina 746.

Informazioni su questa attività

Redistributable Managed File Transfer Agent e Redistributable Managed File Transfer Logger condividono lo stesso ambiente. Una volta creato questo ambiente e impostata la connettività IBM MQ , è possibile creare e avviare il programma di registrazione.

Procedura

1. Assicurarsi che l'ambiente condiviso per Redistributable Managed File Transfer Agent e Redistributable Managed File Transfer Logger sia stato creato come descritto nel passo [“1” a pagina 749](#) e che la connettività IBM MQ sia stata impostata come descritto nel passo [“2” a pagina 750](#) di [“Creazione della configurazione iniziale per Redistributable Managed File Transfer Agent”](#) a pagina 748.
2. Creare un programma di registrazione file utilizzando il comando **fteCreateLogger** .
Ad esempio:

```
fteCreateLogger FILELOGGER -loggerType FILE -loggerQMgr PRMFTDEMO2  
-loggerQMgrHost 9.121.59.233 -loggerQMgrPort 3003 -loggerQMgrChannel SYSTEM.DEF.SVRCONN  
-fileSize 20MB -fileCount 10 -fileLoggerMode CIRCULAR
```

Per ulteriori informazioni sull'utilizzo del comando **fteCreateLogger** , vedi [fteCreateLogger](#).

3. Avviare il programma di registrazione utilizzando il comando **fteStartLogger** .
Per ulteriori informazioni sul comando **fteStartLogger** , consultare [fteStartLogger](#).

Concetti correlati

[“Configurazione di Managed File Transfer”](#) a pagina 744

È possibile configurare le funzioni di Managed File Transfer dopo l'installazione.

[“Opzioni di configurazione MFT su Multiplatforms”](#) a pagina 744

Managed File Transfer fornisce una serie di file delle proprietà che contengono le informazioni chiave sull'impostazione e sono richiesti per l'operazione. Questi file delle proprietà si trovano nella directory di configurazione definita quando è stato installato il prodotto.

Aggiornamento Redistributable Managed File Transfer components

Puoi aggiornare Redistributable Managed File Transfer components scaricando un nuovo Redistributable Managed File Transfer package.

Prima di iniziare

Per informazioni sui termini di licenza redistribuibili per Redistributable Managed File Transfer Agent e Redistributable Managed File Transfer Logger consultare [IBM MQ Componenti redistribuibili](#).

Nota: Advanced Message Security non è supportato con Redistributable Managed File Transfer package.

Informazioni su questa attività

Se Redistributable Managed File Transfer components è già stato installato, è possibile aggiornarlo scaricando un nuovo pacchetto redistribuibile ed estraendo il contenuto nella stessa posizione.

Procedura

1. Scaricare il Pacchetto [agent IBM MQ redistribuibile Managed File Transfer](#) per il proprio sistema operativo da Fix Central.
2. Arrestare tutti gli agent Managed File Transfer e il programma di registrazione attendere il completamento di tutti i comandi Managed File Transfer in esecuzione.
3. Aggiornare i file per l'installazione esistente di Redistributable Managed File Transfer components estraendo il contenuto del nuovo pacchetto redistribuibile scaricato nella stessa directory di quella in cui è già installato Redistributable Managed File Transfer components .

▶ z/OS

Creating an MFT Agent or Logger command data set

You can create a PDSE data set of commands from the Managed File Transfer command template data set for a specific Managed File Transfer Agent or Managed File Transfer Logger for a specific coordination.

About this task

Complete the following steps:

Procedure

1. Make a copy of the MFT command template PDSE library data set SCSQFCMD.
SCSQFCMD must be copied into a new library, for example *prefix.agent.JCL*. You can use an updated version of the SCSQFCMD(BFGCOPY) member with the following replacements:
 - Replace *++supplied-library++* with the fully qualified name of the SCSQFCMD PDSE.
 - ▶ z/OS Replace *++service-library++* with the fully qualified name of the new MFT command PDSE library data set. The *++service-library++* is the output data set for the agent or logger service that is created.
2. For the new MFT command PDSE library data set, edit the member BFGCUSTM, which is a JCL script to customize the commands for the agent or logger. Each variable is specified in the format: *++variable name++*, which you must replace with its required value. For a description of the various JCL variables, see [“z/OS JCL variables” on page 764](#). The BFGSTDIN DD statement defines variables in three categories: Variables, Properties, and Environment. The statement has the following format:

```
[Variables]
variable1=value1
variable2=value2
....
variableN=valueN
[Properties]
property1=property value1
property2=property value2
...
propertyN=property valueN
[Environment]
custom_variable1=value1
custom_variable2=value2
```



```
....  
custom_variableN=valueN
```

Variables define the set of setup and environment variables that are required for each command.

Properties define overrides for the MFT configuration properties. You can add agent and logger properties as required to customize the agent or logger for your environment. For a list of all properties, see [“Configuration properties files” on page 775](#). This facility is provided to save having to access the MFT configuration properties files, which are maintained as z/OS UNIX System Services files.

Environment defines any additionally required custom environment variables.

3. Submit job BFGCUSTM for the new MFT command PDSE library data set. This job generates the set of JCL commands, as new members of the PDSE, appropriate for the agent or logger. For a full list of the commands, see [“z/OS agent and logger command JCL scripts” on page 768](#).

Job BFGCUSTM updates the library containing the JCL which includes a DD statement with DISP=OLD. You must exit the editor after submission to allow the job to execute.

Examine the output job log to check that the JCL script ran successfully. If there are any failures, correct them and submit the BFGCUSTM job again.

The BFGCUSTM JCL script also updates the z/OS UNIX System Services MFT configuration properties files as necessary to keep the files in step. If the configuration defined by the CoordinationQMgr property does not exist, warning messages are output and you must run the generated BFGCFGR and BFGCMCR jobs to create the configuration properties files. You must run BFGAGCR for an agent, and BFGLGCRS for a logger edit. If the specified configuration already exists, the configuration is updated with any properties as defined in the BFTCUSTM JCL script.

Related concepts

[“MFT configuration options on z/OS” on page 746](#)

The Managed File Transfer configuration options on z/OS are the same as the options for distributed platforms.

Related tasks

[“Updating an existing MFT Agent or Logger command data set on z/OS” on page 764](#)

You can update an Managed File Transfer command PDSE library data set that is created from the Managed File Transfer command template data set.

z/OS

Configuring Managed File Transfer for z/OS

Managed File Transfer for z/OS requires customization to enable the component to operate correctly.

About this task

You need to:

1. Edit a PDSE member to specify configuration data
2. Define the coordination queue manager.
3. Define the command queue manager
4. Configure one or more agents
5. Optionally: configure a logger task to store data in Db2

The sequence of tasks you need to perform is detailed in the following topics.

Related concepts

[“Reviewing the MFT configuration” on page 754](#)

You need to review the configuration of your system before you begin.

Related tasks

[Installing IBM MQ Advanced for z/OS](#)

Reviewing the MFT configuration

You need to review the configuration of your system before you begin.

Managed File Transfer (MFT) requires one or more queue managers to act in the following roles for each defined MFT configuration:

- A coordination queue manager, which maintains information on the status of each agent in the configuration published to a topic on the coordinator.
- One or more command or connection queue managers that act as the entry point to the IBM MQ network for MFT commands.
- One or more agent queue managers that provide the communication between an MFT agent and the IBM MQ network.

Each of the above roles can be performed by a separate queue manager, or you can combine the roles, so that, in the simplest configuration, all roles are performed by a single queue manager.

If you are adding a z/OS queue manager to an existing MFT environment you need to define connectivity between the z/OS queue manager and the other queue managers in the configuration. You can achieve this with manually defined transmission queues, or by the use of clustering.

Each MFT agent communicates with a single queue manager. If multiple agents communicate with the same queue manager, then the agent queue manager will have multiple queues defined for each agent:

- SYSTEM.FTE.COMMAND.*agent_name*
- SYSTEM.FTE.DATA.*agent_name*
- SYSTEM.FTE.REPLY.*agent_name*
- SYSTEM.FTE.STATE.*agent_name*
- SYSTEM.FTE.EVENT.*agent_name*
- SYSTEM.FTE.AUTHAGT1.*agent_name*
- SYSTEM.FTE.AUTHTRN1.*agent_name*
- SYSTEM.FTE.AUTHOPS1.*agent_name*
- SYSTEM.FTE.AUTHSCH1.*agent_name*
- SYSTEM.FTE.AUTHMON1.*agent_name*
- SYSTEM.FTE.AUTHADM1.*agent_name*

Note that you can define generic security profiles, where you use a profile such as SYSTEM.FTE.COMMAND.* , or you can define specific profiles for each agent.

Related concepts

[“Prima di avviare la configurazione di MFT per z/OS” on page 754](#)

La configurazione Managed File Transfer (MFT) utilizza i file in dataset z/OS UNIX System Services (z/OS UNIX) e PDSE.

Related reference

[MFT system queues and the system topic](#)

Prima di avviare la configurazione di MFT per z/OS

La configurazione Managed File Transfer (MFT) utilizza i file in dataset z/OS UNIX System Services (z/OS UNIX) e PDSE.

La maggior parte della configurazione e dell'operazione viene eseguita utilizzando JCL da un PDSE ed è necessario avere dimestichezza con le operazioni in un ambiente z/OS UNIX .

È possibile accedere a OMVS da ISPF oppure è possibile utilizzare una sessione di tipo Telnet utilizzando i comandi sulla workstation, ad esempio Telnet Putty o SSH.

Se si utilizza OMVS da ISPF , è possibile utilizzare l'editor ISPF standard e sfogliare i comandi **oedit** e **obrowse**.

È necessario avere dimestichezza con i seguenti comandi z/OS UNIX

<i>Tabella 41. Comandi z/OS UNIX comuni</i>	
Comando	Funzione
percorso chmod xxx	Modificare le autorizzazioni di accesso ai file.
percorso df -k	Riporta la quantità di spazio libero rimanente nel file system. -k riporta lo spazio libero in KB.
Percorso du -kt	Riporta le dimensioni delle directory nel percorso. Dimensione riportata in KB.
trova percorso -name xxx	Ricerca il file denominato xxxx nella directory del percorso. xxx è sensibile al maiuscolo / minuscolo e può essere simile a *zzz.
directory ls -ltrd	Elenca le informazioni sulla directory specificata piuttosto che i file nella directory.
percorso ls -ltr	Elenca le informazioni sui file nel percorso.
nome file obrowse	Sfoggia il nome file.
nome file oedit	Modificare un file in OMVS.

Esaminare gli elementi nella seguente tabella e completare la tabella con le voci appropriate per la propria azienda. Questi valori sono necessari quando si modifica un membro BFGCUSTM.

<i>Tabella 42. Parametri necessari per il membro BFGCUSTM</i>		
Nome	Dati di esempio	Commenti
ADMIN_JOB1		Scheda di lavoro. Tutti i lavori vengono generati con la stessa scheda JCL.
armELEMENT	Se viene utilizzato ARM, utilizzare il valore ARM ELEMENT specificato nella politica ARM per questo agent o programma di registrazione. Se ARM non viene utilizzato, impostare questo parametro su uno spazio vuoto; ad esempio, armELEMENT=	
armELEMENTYPE	Se si sta utilizzando ARM, utilizzare ARM ELEMENTYPE specificato nella politica ARM. Ad esempio, armELEMENTYPE= SYSBFGAG per un agente o armELEMENTYPE= SYSBFGLG per un programma di registrazione. Se ARM non viene utilizzato, impostare questo parametro su uno spazio vuoto; ad esempio, armELEMENTYPE=	
DATI_BFG		Completare come necessario
NOME_GRUPPO_BF	MQM	

<i>Tabella 42. Parametri necessari per il membro BFGCUSTM (Continua)</i>		
Nome	Dati di esempio	Commenti
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_JVM_PROPERTIES		Completare come necessario
BFG_PROD	/mqm/V9R2M0/mqft	Il percorso completo della directory mqft nella directory IBM MQ for z/OS UNIX System Services Components .
BFG_WTO	Sì	Per ottenere un messaggio MFT sul syslog.
CLEAN_AGENT_PROPS	-trs	Questo parametro specifica le opzioni che verranno utilizzate per ripulire un agent quando viene eseguito un membro BFGAGCL. Per ulteriori informazioni sui valori validi per questo parametro, consultare fteCleanAgent: cleanup di MFT Agent .
coordinationQMgr	MQPV	Configurazione obbligatoria
PERCORSO CREDENTIAL_PATH		Utilizzato nella migrazione
Db2_HLQ	SYS2.Db2.V10	
PERCORSO_PROP_DB		Utilizzato nella migrazione
FTE_CONFIG		Utilizzato nella migrazione
JOBCARD1		Questa è la scheda di lavoro per le attività di lunga durata, gli agent e i logger.
LIBRERIA	SCEN.FTE.JCL	Nome del PDSE MFT. È necessaria una copia per ogni attività dell'agent o del programma di registrazione.
MQ_HLQ	Il qualificatore di alto livello per i dataset IBM MQ . Ad esempio, MQM.V920	
MQ_LANG	E	
PERCORSO MQ	/mqm/V9R2M0	Il percorso completo della directory per l'installazione di IBM MQ for z/OS UNIX System Services Components.
NOME	AGENT1	
CLASSE_OUTPUT	*	
PERCORSO	bin:/usr/bin:/usr/sbin	

Tabella 42. Parametri necessari per il membro BFGCUSTM (Continua)

Nome	Dati di esempio	Commenti
productId	AVANZATOVUE	Questo parametro viene utilizzato per impostare il tipo di prodotto per cui deve essere registrato l'uso di Managed File Transfer . Per informazioni sui valori validi per questo parametro, consultare fteSetProductId: set z/OS SCRT recording product id.
QMGR	MQPV	
service_type	AGENT o LOGGER	
TMPDIR	/tmp	Percorso z/OS UNIX accessibile in lettura e scrittura per i file temporanei.

Inoltre, è necessario rivedere le seguenti variabili e fornire i valori, se necessario:

- Host coordinationQMgr=
- coordinationQMgrPorta =
- Canale coordinationQMgr=
- connectionQMgr=
- Host connectionQMgr=
- connectionQMgrPorta =
- Canale connectionQMgr=

Queste proprietà sono comuni a AGENT o LOGGER.

Nota: Host, Porta e Canale sono richiesti per la connessione client, ma devono rimanere vuoti per una connessione di bind sulla macchina locale.

Concetti correlati

[“Items to check” a pagina 757](#)

Ensure that you have enough disk space, a directory for storing data, and that the requisite files exist.

[“Editing member BFGCUSTM” a pagina 760](#)

You must edit member BFGCUSTM, and enter the values for the parameters that your enterprise uses, before you run the job.

Items to check

Ensure that you have enough disk space, a directory for storing data, and that the requisite files exist.

Check you have enough disk space

Check that you have enough disk space available on the file system where you are going to store the configuration specific files.

If an agent trace is enabled then by default it can use 100 MB of disk space.

The configuration files themselves are small, only a few KB in size.

If you are planing on using two agents and a logger then you need at least 300 MB. You can use the command **df -k path**, where path is the location of the installation specific files. This gives the available and total space in KB.

300 MB is 307,200 KB so you should allow for at least 310,000 KB

Create and check the directory for storing Managed File Transfer data

You need a directory for storing the Managed File Transfer (MFT) data.

Check you have enough space in the file system `df -k /var`. This file system should have at least 310,000 KB available.

If you have not created this file system, use the `mkdir` command; for example `mkdir /var/mft`.

Display what permissions users have on this directory, using the command `ls -ltrd /var/mft`.

If the owner or group is not correct, use the command `chown owner:group /var/mft`.

If permissions for the group are not correct, use the following command to give the owner and the group read, write, and execute permissions. Note that the following command also gives all users read and execute permissions `chmod 775 /var/mft`.

Check the files exist and you have access to them

Use the `ls -ltr` command for the files you will be using during customization. For example:

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

gives

```
drwxr-xr-x 4 SYSTASK TSouser 8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

where the `drwxr-xr-x` means

d

This is a directory.

rwX

The owner `SYSTASK` has read, write and execute access to the directory.

r-x

People in the group `TSouser` can read and execute files in the directory.

r-x

Universal access, that is, anyone can read or execute files in the directory.

Check the files specified in:

<i>Table 43. Access required by users to specific files</i>	
Path	Access required by users doing the configuration
BFG_JAVA_HOME	Read and execute
/tmp	Read and write
BFG_PROD	Read
BFG_DATA	Write
MQ_PATH	Read

Related concepts

[“Prima di avviare la configurazione di MFT per z/OS” on page 754](#)

La configurazione Managed File Transfer (MFT) utilizza i file in dataset z/OS UNIX System Services (z/OS UNIX) e PDSE.

[“Common MFT for z/OS configurations” on page 759](#)

An overview of the different Managed File Transfer configurations

Common MFT for z/OS configurations

An overview of the different Managed File Transfer configurations

Managed File Transfer uses agents attached to a queue manager for transferring data.

MFT può utilizzare più gestori code:

- Uno o più gestori code per il trasferimento dei dati.
- Un gestore code di comandi che emette le richieste. Ad esempio, una richiesta per avviare un trasferimento viene inviata a questo gestore code e i comandi associati vengono instradati agli agent MFT.
- Un gestore code di coordinamento che gestisce il lavoro.

Esistono tre configurazioni comuni di Managed File Transfer (MFT):

1. Un singolo gestore code con uno o più agent che utilizzano connessioni locali. Potrebbe essere utilizzato per inserire il contenuto di un dataset in code di IBM MQ.
2. Un singolo gestore code con un client MFT su una macchina distribuita utilizzando bind del client.
3. Due gestori code connessi mediante canali e uno o più agent su ciascuna macchina. Questi agent possono essere bind del client o locali.

Tenere presente i seguenti aspetti:

1. MFT è scritto in Java, con alcuni script di shell e JCL per configurare e utilizzare MFT.
2. È possibile registrare lo stato e l'attività di Db2 e tale registrazione può essere memorizzata nelle tabelle Db2.
3. La persona che configura MFT deve avere dimestichezza con z/OS UNIX System Services (z/OS UNIX).
Ad esempio:
 - La struttura di directory con file con nomi come `/u/userID/myfile.txt2`
 - Comandi z/OS UNIX, quali:
 - cd** (modifica directory)
 - ls** (elenca)
 - chmod** (modifica autorizzazioni file)
 - chown** (modifica la proprietà dei file o i gruppi che possono accedere al file o alla directory)
4. I seguenti prodotti sono richiesti in z/OS UNIX per poter configurare ed eseguire MFT:
 - Java; ad esempio, `/java/java71_bit64_GA/J7.1_64/`
 - IBM MQ V920, ad esempio `/mqm/V9R2M0`.
 - Le librerie JDBC Db2, se si desidera utilizzare Db2 per lo stato e la cronologia, ad esempio `/db2/db2v12/jdbc/lib`

You need a coordination queue manager. However, you can use the same queue manager to run agents, to process commands, and for coordination. If you are using multiple queue managers, you must pick one to act as the coordinator.

Check your IBM MQ connectivity

If you have an existing MFT coordinator queue manager, you need connectivity between the queue manager where you are doing the configuration, and the coordinating and command queue managers.

Copy SCSQFCMD to create a JCL library

You need to create a JCL library for each agent and logger. The JCL contains the configuration and jobs used to create and run the agent or logger.

For each agent and logger create a copy of the IBM supplied SCSQFCMD library by editing and running the BFGCOPY member.

This library is used to define the configuration for the agent or logger and, after customization, contains jobs that can be used to create the required Managed File Transfer configuration and agent or logger.

You create member BFGCUSTM as part of this process.

Note: If you are familiar with z/OS UNIX commands, you can configure z/OS with the same commands that you use on other platforms.

Related concepts

[“Common MFT for z/OS configurations” on page 759](#)

An overview of the different Managed File Transfer configurations

[“Editing member BFGCUSTM” on page 760](#)

You must edit member BFGCUSTM, and enter the values for the parameters that your enterprise uses, before you run the job.

Editing member BFGCUSTM

You must edit member BFGCUSTM, and enter the values for the parameters that your enterprise uses, before you run the job.

See [Parameters needed for member BFGCUSTM](#), for a list of the parameters requiring specific values.

Inoltre, è necessario rivedere le seguenti variabili e fornire i valori, se necessario:

- Host coordinationQMgr=
- coordinationQMgrPorta =
- Canale coordinationQMgr=
- connectionQMgr=
- Host connectionQMgr=
- connectionQMgrPorta =
- Canale connectionQMgr=

Queste proprietà sono comuni a AGENT o LOGGER.

Note: Host, Porta e Canale sono richiesti per la connessione client, ma devono rimanere vuoti per una connessione di bind sulla macchina locale.

If this is the first queue manager in your Managed File Transfer environment, and you want to use the same queue manager for coordination, commands, and running agents, set the values to the local queue manager name.

```
coordinationQMgr=MQPV
connectionQMgr=MQPV
```

where MQPV is your local queue manager name.

Submit the job, which updates the PDSE, and creates a directory structure under the specified path.

Note that this job requires exclusive use, so you need to stop using the PSDE while the job runs.

Tip: Whenever you submit job BFGCUSTM, the job replaces all the JCL files. You should rename each member you change.

Related concepts

[“Prima di avviare la configurazione di MFT per z/OS” on page 754](#)

La configurazione Managed File Transfer (MFT) utilizza i file in dataset z/OS UNIX System Services (z/OS UNIX) e PDSE.

[“Creating an agent” on page 762](#)

You need to copy the PDSE to make the agent-specific PDSE, for example *user.MFT.AGENT1*. Copy the PDSE from a previous agent or logger configuration, if they exist. If this is your first configuration, copy the PDSE supplied with MFT.

Defining the coordination queue manager

Managed File Transfer requires a queue manager to be created that acts as the coordination queue manager.

Depending on the configuration that you have chosen, this queue manager is on the local MVS system, or on another machine. In the former case, the connections to it are bindings connections and in the latter case, they are client connections.

After you have run the configuration step successfully there are configured members in the PDSE.

Member BFGCFR defines the coordination queue manager, and this job:

1. Creates a directory structure in the Managed File Transfer (MFT) directory, and creates configuration files.
2. Runs CSQUTIL to define IBM MQ resources.

If the coordination queue manager is on a remote machine then this job step fails.

Member BCFCFR creates files in z/OS UNIX System Services and creates MQ definitions. This job:

1. Creates an MFT topic,
2. Creates an MFT queue
3. Alters `NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` to be `NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)`
4. Performs `ALTER QMGR PSMODE(ENABLED)`

A `DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` command is issued before doing the alter. If your NAMLIST is not the default, you should alter your name list to add SYSTEM.FTE to your namelist

Rename member BCFCFR with your own prefix, for example, CCPCFR, because re customizing this file replaces it.

Edit this renamed member by inserting the name of your credentials file. For example:

```
%BFGCMD CMD=fteSetupCoordination +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

Save and submit the job. Note that if you need to resubmit the job, you need to add the `-f` option.

When this job runs it lists the IBM MQ resources it creates. You need to protect these resources.

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST
* Value prior to alteration:
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)
* Altering PSMODE. Value prior to alteration:
DISPLAY QMGR PSMODE
ALTER QMGR PSMODE(ENABLED)
```

Related tasks

[“Defining the command queue manager” on page 761](#)

You can either use the same queue manager as the coordination and command queue managers, or create a new command queue manager.

Defining the command queue manager

You can either use the same queue manager as the coordination and command queue managers, or create a new command queue manager.

About this task

You must have a command queue manager, however, you can use the same queue manager for the coordination and command queue managers. Otherwise, you need to create a new command queue manager. This can be on the same machine as the coordination queue manager, but does not have to be.

Procedure

1. Rename member BFGCMCR with your own prefix, for example, CCPCMCR.
You must rename BFGCMCR because re-customizing this file replaces it.
2. Edit the renamed member by inserting the name of your credentials file.
For example:

```
%BFGCMD CMD=fteSetupCommands +  
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
```

3. Save and submit the job.
Note that if you need to resubmit the job, you need to add the *-f* option.
This queue manager is used for commands such as **ftePingAgent**.
4. Review this member, submit it, and review the output.

What to do next

See [“Creating an agent” on page 762](#) for information on how you create an agent.

Related concepts

[“Defining the coordination queue manager” on page 761](#)

Managed File Transfer requires a queue manager to be created that acts as the coordination queue manager.

Related tasks

[Configuring MQMFTCredentials.xml](#)

Related reference

[MFT credentials file format](#)

Creating an agent

You need to copy the PDSE to make the agent-specific PDSE, for example *user.MFT.AGENT1*. Copy the PDSE from a previous agent or logger configuration, if they exist. If this is your first configuration, copy the PDSE supplied with MFT.

Review member BFGCUSTM and if you need to use a different credentials file, create one.

Much of the content remains the same from the customization detailed in [“Editing member BFGCUSTM” on page 760](#).

You need to change:

- //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- LIBRARY to match the agent PDSE
- SERVICE_TYPE=AGENT
- NAME to be the name of the agent (matching the PDSE) JOBCARD
- Change BFG_JVM_PROPERTIES="-Xmx1024M"

Submit this job, remembering that the job requires exclusive access to the data set.

The jobs for the agent all have names of the form *BFGAG**

Rename member *BFGAGCR*. This job updates files in the Managed File Transfer directory and uses CSQUTIL to create agent specific queues in the local queue manager. Specify the name of your credentials file, for example, `-credentialsFile // 'SCEN.FTE.JCL.VB(CREDOLD)`. If you do not specify the name, the job to start the agent does not use a credentials file.

Check the output to ensure that the process has run successfully.

Tip: Copy the path name of the *agent.properties* file from the output of the job to a member in the PDSE for the agent.

For example, copy `/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties` into member AGENT.

This is useful if you need to display the properties file, and add the line `/u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs`.

This is where trace files are stored.

Related concepts

[“Defining the coordination queue manager” on page 761](#)

Managed File Transfer requires a queue manager to be created that acts as the coordination queue manager.

[“Using the agent” on page 763](#)

How you use various commands to ensure that the agent is working correctly.

Related tasks

[“Defining the command queue manager” on page 761](#)

You can either use the same queue manager as the coordination and command queue managers, or create a new command queue manager.

Using the agent

How you use various commands to ensure that the agent is working correctly.

Start the agent

Rename member BFGAGST, review the member, and submit the job.

If this works you receive message BFGAG0059I: The agent has been successfully started.

Display the active agent(s)

Rename member BFGAGLI, review the member and submit the job which uses the coordinating queue manager.

You must resolve any connectivity problems

Ping the agent to check it is working

Rename member BFGAGPI, review the member and submit the job which uses the command queue manager.

You must resolve any connectivity problems

Carry out a test transfer

See [“Performing a verification transfer” on page 770](#) for further information.

Stop the agent

Rename member BFGAGSP, review the member and submit the job.

Restart the agent using the member BFGAGST.

Related concepts

[“Creating an agent” on page 762](#)

You need to copy the PDSE to make the agent-specific PDSE, for example *user.MFT.AGENT1*. Copy the PDSE from a previous agent or logger configuration, if they exist. If this is your first configuration, copy the PDSE supplied with MFT.

Updating an existing MFT Agent or Logger command data set on z/OS

You can update an Managed File Transfer command PDSE library data set that is created from the Managed File Transfer command template data set.

Procedure

1. Edit the BFGCUSTM JCL script member and update variables and properties in the BFGSTDIN DD statement.

If you want to remove a property that was previously defined, set its value to blank, instead of removing the entry. When the BFGCUSTM JCL script is run, the specified properties are applied as an update to the actual agent and logger z/OS UNIX System Services properties files; setting a property to a blank value indicates that the property is to be removed

2. Submit job BFGCUSTM. This job generates the set of JCL commands again, appropriate for the agent or logger. For a full list of the commands, see [“z/OS agent and logger command JCL scripts” on page 768](#). Examine the output job log to check that the JCL script ran successfully. If there are any failures, correct them and submit the BFGCUSTM job again.

Results

You can modify the generated JCL scripts and add your own logic. However, be careful when you run BFGCUSTM again because you might overwrite the custom logic.

Related concepts

[“MFT configuration options on z/OS” on page 746](#)

The Managed File Transfer configuration options on z/OS are the same as the options for distributed platforms.

Related tasks

[“Creating an MFT Agent or Logger command data set” on page 752](#)

You can create a PDSE data set of commands from the Managed File Transfer command template data set for a specific Managed File Transfer Agent or Managed File Transfer Logger for a specific coordination.

z/OS JCL variables

You can use substitution values, JCL variables, and configuration properties in the BFGCUSTM script.

The following table lists the substitution values for the BFGCUSTM JCL script in an MFT command PDSE library data set. You must replace these substitution values with suitable values before you submit the BFGCUSTM job.

Substitution variable	Value
++library++	The data set name of the containing MFT command PDSE library.
++bfg_java_home++	The location of your Java installation.

Table 44. Substitution values for the BFGCUSTM JCL script (continued)

Substitution variable	Value
++mq_path++	The path to the IBM MQ for z/OS UNIX System Services Components directory. For example, /mqm/V9R2M0. This is used to give the full path to the MFT installation, for example, /mqm/V9R2M0/mqft.

The following table describes the environment variables for the BFGSTDIN DD statement for the BFGCUSTM JCL script, in an MFT command PDSE library data set (in the [Variables] section). You must replace all variables that are specified with substitution values (that is, values enclosed in two plus signs, ++), with suitable values before you submit the BFGCUSTM job.

Table 45. Environment variables

Environment variable	Value
LIBRARY	The data set name of the containing MFT command PDSE library.
TMPDIR	z/OS UNIX System Services directory for temporary files.
BFG_PROD	The full path to the mqft directory under the IBM MQ for z/OS UNIX System Services Components directory; For example: /mqm/V9R2M0/mqft.
BFG_DATA	The location of the data directory for Managed File Transfer for z/OS, which is the path to <i>DATA_DIR</i> .
BFG_JAVA_HOME	The location of your Java installation.
BFG_JVM_PROPERTIES	Optional. Sets a value for the BFG_JVM_PROPERTIES environment variable. These properties are passed to the Java virtual machine.

Table 45. Environment variables (continued)

Environment variable	Value
BFG_GROUP_NAME	<p>The mqm file group is typically associated with MFT configuration data files and commands. Consequently, all users who are members in the mqm group can access and make changes to the MFT configuration. For more information, see File system permissions for MFT in IBM MQ.</p> <p>For a z/OS system, a file group is a z/OS UNIX System Services (z/OS UNIX) filesystem entity, and the mqm file group is not necessarily defined. You can associate a z/OS UNIX filesystem group for MFT configuration data files by using the BFG_GROUP_NAME environment variable. For example, at the z/OS UNIX shell prompt use:</p> <pre data-bbox="862 680 1472 758">export BFG_GROUP_NAME=FTEGB</pre> <p>which defines a group <i>FTEGB</i> to be associated with any subsequently created configuration files for the current z/OS UNIX session.</p> <p>You can set BFG_GROUP_NAME to a blank value, or remove it.</p> <p>Note: When running BFGCUSTM for the first time, if the MFT configuration is to be used by multiple user IDs, it is important that BFG_GROUP_NAME is set to a group accessible to all required user ID's. If BFGCUSTM is run again, then BFG_GROUP_NAME must not be changed (otherwise, the z/OS UNIX group file permissions for all files and directories in the directory referenced by BFG_DATA must also be changed to reflect the new BFG_GROUP_NAME setting).</p>
BFG_WTO	<p>z/OS logging is enabled when BFG_WTO is set to YES, ON, or TRUE. This controls whether messages that are written to the agent event log are also written to the z/OS operator log facility, which allows easier access for automation products when you run an agent from JCL. The routing code is Programmer Information (11) and the descriptor code is Informational (12).</p>
SERVICE_TYPE	<p>Specifies whether the MFT command library is for an agent or logger. The valid values are AGENT or LOGGER.</p>
NAME	<p>The name of the agent or logger for the SERVICE_TYPE value.</p>
QMGR	<p>The name of the local queue manager that is associated with the agent or logger for the SERVICE_TYPE value.</p>

<i>Table 45. Environment variables (continued)</i>	
Environment variable	Value
OUTPUT_CLASS	The output class for SYSOUT data sets. Defaults to * which requests the same output class as the MSGCLASS parameter from the job statement.
MQ_PATH	The path to the IBM MQ for z/OS UNIX Components directory.
MQ_HLQ	The high-level qualifier for IBM MQ data sets.
MQ_LANG	The language that is required.
DB2_HLQ	Optional. High-level qualifier for Db2 data sets.
JOBCARD1	Header line 1 for a JCL command job.
JOBCARD2	Header line 2 for a JCL command job.
JOBCARD3	Header line 3 for a JCL command job.
ADMIN_JOB1	Header line 1 for an admin job.
ADMIN_JOB2	Header line 2 for an admin job.
ADMIN_JOB3	Header line 3 for an admin job.
FTE_CONFIG	Existing MFT configuration for migration. Set to a blank value if migration is not required.
CREDENTIAL_PATH	Path to credentials file for migration, for example /u/user1/agent3. Required for migration commands BFGAGMG and BFGLGMG JCL scripts only. Set to a blank value if migration is not required. Note also that
DB_PROPS_PATH	Specifies the database logger properties file for migration. This option is required only if the properties file does not use the following default name and path: config_directory/coordination_qmgr/ databaselogger.properties. Set to a blank value if migration is not required.

The following table describes the mandatory MFT configuration properties for the BFGSTDIN DD statement for the BFGCUSTM JCL script in an MFT command PDSE library data set. You must replace properties specified with substitution values (that is, values enclosed in two plus signs, ++) with a suitable non-blank value before you submit the BFGCUSTM job. These properties define overrides for the MFT configuration properties. You can add agent and logger properties to customize agents or loggers for your environment. For a list of all properties, see “Configuration properties files” on page 775.

<i>Table 46. Mandatory configuration properties for the BFGSTDIN DD statement</i>	
Property	Value
coordinationQMGr	The name of the coordination queue manager for the configuration that the agent or logger is associated with.
coordinationQMGrHost	Optional. Host name of the system that the coordination queue manager is running on. If you leave the value for this property blank, a bindings mode connection is assumed.

Table 46. Mandatory configuration properties for the BFGSTDIN DD statement (continued)

Property	Value
coordinationQMGrPort	Optional. Port number that the coordination queue manager is listening on. This parameter is used only if you also specify a non-blank value for the coordinationQMGrHost property.
coordinationQMGrChannel	Optional. Channel to use to connect to the coordination queue manager. This parameter is used only if you also specify a non-blank value for the coordinationQMGrHost property.
connectionQMGr	The name of the command queue manager for the configuration that the agent or logger is associated with.
connectionQMGrHost	Optional. Host name of the system that the command queue manager is running on. If you leave the value for this property blank, a bindings mode connection is assumed.
connectionQMGrPort	Optional. Port number that the command queue manager is listening on. This parameter is used only if you also specify a non-blank value for the connectionQMGrHost property.
connectionQMGrChannel	Optional. Channel to use to connect to the command queue manager. This parameter is used only if you also specify a non-blank value for the connectionQMGrHost property.

z/OS agent and logger command JCL scripts

The set of JCL commands available in an MFT command PDSE library data set.

Table 47. JCL commands available in an MFT command PDSE library data set

Member	Description or fte command line command
BFGCOPY	Job to create a copy of this library
BFGCUSTM	Job to customize this library for agent or logger
BFGZCFRC	fteSetupCoordination
BFGZCMCR	fteSetupCommands: create the MFT command.properties file
BFGZAGCR	fteCreateAgent . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGLGCRS	fteCreateLogger . Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGZAGST	fteStartAgent . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGAGSTP	fteStartAgent procedure. Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZAGPI	ftePingAgent . Created only when you set the SERVICE_TYPE variable to AGENT.

Table 47. JCL commands available in an MFT command PDSE library data set (continued)

Member	Description or fte command line command
BFGZAGSP	<code>fteStopAgent</code> . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZLGST	<code>fteStartLogger</code> . Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGLGSTP	fteStartLogger procedure. Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGZLGSP	<code>fteStopLogger</code> . Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGZAGSH	<code>fteShowAgentDetails</code> . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZLGSH	<code>fteShowLoggerDetails</code> . Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGZCFDF	<code>fteChangeDefaultConfigurationOptions</code>
BFGZAGCL	<code>fteCleanAgent</code> . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZAGDE	<code>fteDeleteAgent</code> . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZLGDE	<code>fteDeleteLogger</code> . Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGZPRSH	<code>fteDisplayVersion</code>
BFGZAGLI	<code>fteListAgents</code> . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZMCLI	<code>fteListMonitors</code>
BFGZSTLI	<code>fteListScheduledTransfers</code>
BFGZTMLI	<code>fteListTemplates</code>
BFGXCROB	fteObfuscate sample
BFGZRAS	fteRAS
BFGZAGTC	<code>fteSetAgentTraceLevel</code> . Created only when you set the SERVICE_TYPE variable to AGENT.
BFGZLGTC	<code>fteSetLoggerTraceLevel</code> . Created only when you set the SERVICE_TYPE variable to LOGGER.
BFGXPRAN	fteAnt sample
BFGXTRCA	fteCancelTransfer sample
BFGXMNCR	fteCreateMonitor sample
BFGXTMCR	fteCreateTemplate sample
BFGXTRCR	fteCreateTransfer sample
BFGXMNDE	fteDeleteMonitor sample
BFGXSTDE	fteDeleteScheduledTransfer sample
BFGXTMDE	fteDeleteTemplate sample

Notes:

- The JCL, for commands that create MQSC or reference delete scripts, asks you to run a script, but the script has already been run by the job.
- BFGZRAS creates the BFGRAS member when the BGCUSTM job is run.

Performing a verification transfer

How you carry out a transfer to check that the product is working correctly.

Rename and edit member BFGTRCRS.

1. Add a /* before the %BFGCMD CMD=fteCreateTransfer -h
2. Remove the other comments in the member.
3. Specify the current agent name for -sa and -da
4. Save the JCL
5. Submit the JCL

This JCL connects to the command queue manager.

Configurazione di un'attività di registrazione

L'attività di registrazione deve essere eseguita sulla stessa immagine del gestore code di coordinamento. È possibile accedere a Db2.

Creazione di un'attività di registrazione

Copiare il PDSE per rendere il PDSE specifico del programma di registrazione. Ad esempio, user.MFT.LOGGER.

Se è necessario utilizzare un file di credenziali diverso, crearne uno. Consultare [Configuring MQMFTCredentials.xml on z/OS](#).

Membro di riesame BFGCUSTM. Si noti che la maggior parte del contenuto rimane lo stesso della personalizzazione precedente.

Tuttavia, è necessario:

- Modificare // SYSEXEC DD DSN=SCEN.FTE.JCL....
- Modificare LIBRARY in modo che corrisponda al PDSE agent
- Modificare QMGR con il nome del gestore code di coordinamento
- Rendi TIPO_SERVIZIO=LOGGER
- Modificare NAME in modo che sia il nome del programma di registrazione (corrispondente al PDSE)
- Rivedere JOBCARD e modificare il nome lavoro in modo che sia diverso dai nomi lavoro degli agent.
- Esaminare BFG_JVM_PROPERTIES = "-Xmx1024M"

Se si utilizza il programma di registrazione Db2 , è utile creare un file, in modo da poter catturare le tracce Db2 per identificare i problemi Db2 .

Il nome del file viene specificato nelle proprietà JVM, in cui il file delle proprietà di traccia JDBC ha il contenuto come

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

Imposta due proprietà JVM

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties  
-Ddb2.jcc.ssid=DBCA
```

dove `/u/.../sql.properties` è il nome del file delle proprietà di traccia Db2 e `DBCA` è il nome del sottosistema Db2 .

Inoltre questo lavoro, notando che il lavoro richiede accesso esclusivo al dataset. I lavori per l'agent hanno tutti nomi come `BFGLG*`.

Registrazione nei file

Per ulteriori informazioni sulla registrazione in Db2, consultare [“Creazione di un'attività di registrazione, quando si accede a Db2”](#) a pagina 772

Ridenominare il membro BFGLCRS. Questo lavoro aggiorna i file nella directory Managed File Transfer (MFT) e utilizza CSQUTIL per creare code specifiche dell'agente nel gestore code locale.

Il file originale ha il comando `%BFGCMD CMD=fteCreateLogger -h` che elenca la sintassi del comando.

To create the logger task comment out the `%BFGCMD CMD=fteCreateLogger -h` by putting `/*` in front of the statement, making sure that column one is blank.

Rimuovere i commenti dal secondo comando e configurare le istruzioni. Ad esempio:

```
%BFGCMD CMD=fteCreateLogger  +  
-p MQPH  +  
-loggerQMgr MQPH  +  
-loggerType FILE  +  
-fileLoggerMode circular  +  
-fileSize 5MB  +  
-fileCount 5  +  
-p MQPH  +  
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'  
LOGGER
```

Controllare l'output per verificare che sia stato elaborato correttamente.

Suggerimento: Copiare il nome percorso del file `logger.properties` dall'emissione del lavoro ad un membro nel PDSE dell'agente.

Ad esempio, copia nel membro APATH

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

Ciò è utile se è necessario visualizzare il file delle proprietà.

Aggiungere la directory a questo file:

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

Se si sta eseguendo l'accesso al file, i file di log vengono memorizzati in questa directory, ad esempio `LOGGER0-20140522123654897.log`.

I file di traccia si trovano nella sottodirectory di log, ad esempio

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

È ora possibile [avviare l'attività di registrazione](#).

Creazione di un'attività di registrazione, quando si accede a Db2

Ridenominare il membro BFGLGCRS.

Questo lavoro aggiorna i file nella directory MFT e utilizza CSQUTIL per creare code specifiche dell'agente nel gestore code locale.

Devi sapere:

Nome Db2	Esempio
-dbName databaseName	È possibile ottenere questo valore dal valore di posizione nel messaggio DSNL004I per il proprio sottosistema Db2
-dbDriver filePath	Ad esempio /db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	Ad esempio /db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

Modifica il file. Il file originale ha il comando %BFGCMD CMD=fteCreateLogger -h che elenca la sintassi del comando.

Rimuovere i commenti dal secondo comando e configurare le istruzioni. Per esempio

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerQMgr MQPH +
-loggerType DATABASE +
-dbType DB2 +
-databaseName DSNDBCP +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar +
-dbLib /db2/db2v10/jdbc/lib/ +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

To create the logger task comment out the %BFGCMD CMD=fteCreateLogger -h by putting /* in front of the statement, making sure that column one is blank.

Inoltre il lavoro e controllare l'output per verificare che sia stato elaborato correttamente.

Suggerimento: Copiare il nome percorso del file logger.properties dall'emissione del lavoro a un membro nel PDSE degli agenti.

Ad esempio, copiare nel membro APATH:

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

Ciò è utile se è necessario visualizzare il file delle proprietà

I file di traccia si trovano nella sottodirectory di log, ad esempio:

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

Creazione di tabelle Db2

È necessario creare le tabelle Db2. Le definizioni si trovano nel z/OS UNIX System Services file mqft/sql/ftelog_tables_zos.sql.

Creare un membro Db2 nel PDSE. Modificare questo membro e usare il comando COPY sulla riga comandi. Copiare dal file di definizione z/OS UNIX System Services.

Poiché i requisiti specifici del sito possono variare notevolmente, questo file specifica solo le strutture di base delle tabelle e un tablespace in cui si trovano.

Lo spazio tabella viene specificato, mediante lo script SQL, per garantire che venga creato utilizzando un pool di buffer con una dimensione di pagina sufficiente a contenere le righe di tabelle più grandi possibili. Notare che gli attributi come le ubicazioni LOB e così via non vengono specificati.

L'amministratore del database potrebbe voler modificare una copia di questo file, per definire questi attributi relativi alle prestazioni.

Questo file assume anche un nome schema predefinito di FTELOG, un nome tablespace predefinito di FTELOGTSe un nome database FTELOGDB. È possibile modificare questi nomi se necessario, in modo che corrispondano a un database esistente e a qualsiasi convenzione di denominazione locale, seguendo il processo descritto nei commenti all'inizio del file.

Importante: Utilizzare funzioni in linea come **SPUFI** per eseguire i comandi, poiché nel file sono presenti commenti e i programmi batch come **DSNTINAD** non accettano commenti.

Per ulteriori informazioni, consultare [Esecuzione di SQL utilizzando SPUFI](#). Inoltre, CSQ45STB in SCSQPROC dispone di un JCL di esempio che è possibile personalizzare per eseguire i comandi Db2 SELECT.

Avvio dell'attività del logger

Ridenominare, esaminare e inoltrare il membro BFGLGST. Si dovrebbe ricevere il messaggio BFGDB0023I: il programma di registrazione ha completato attività di avvio ed è ora in esecuzione.

Operazioni logger

Per visualizzare lo stato del programma di registrazione, ridenominare, esaminare e inoltrare il membro BFGLGSH

Per arrestare il programma di registrazione, rinominare, rivedere e inoltrare il membro BFGLGSP.

Environment variables for MFT on z/OS

If you are running commands direct from the z/OS UNIX System Services (z/OS UNIX) environment, or your own JCL scripts, after customization and configuration you must set a number of environment variables before running the configuration and administration scripts provided by Managed File Transfer. You must set these variables for each user and in each environment that the scripts will be invoked from.

To avoid conflicts with other products, you can choose to create a `.wmqfterc` script in your home directory. The `.wmqfterc` script is then invoked by each of the Managed File Transfer scripts and you can use this script to provide custom environment settings for Managed File Transfer.

There is also one optional environment variable, `BFG_WTO`, that you can set to send messages to the operator log when running agents from JCL.

<i>Table 49. Required z/OS environment variables</i>	
Environment variable	Value
BFG_JAVA_HOME	The location of your Java installation. For more information about the levels of Java supported, see System Requirements for IBM MQ .
BFG_DATA	The location of the data directory for Managed File Transfer for z/OS. This is the path to <code>DATA_DIR</code> .

Table 49. Required z/OS environment variables (continued)

Environment variable	Value
STEPLIB	<p>Must include the following IBM MQ data sets:</p> <ul style="list-style-type: none"> • SCSQAUTH • SCSQANLE • SCSQLOAD <p>If you want to run the database logger component on a z/OS system, STEPLIB must also include the following Db2 data sets in the order shown:</p> <ul style="list-style-type: none"> • SDSNEXIT • SDSNLOD2 • SDSNLOAD

The following is an example .profile that correctly configures the environment variables for Managed File Transfer:

```
STEPLIB=MQM.V920.SCSQAUTH:MQM.V920.SCSQANLE:MQM.V920.SCSQLOAD
PATH=/u/ftuser/bin:/u/ftuser/J7.0/bin:/bin:/usr/bin:/u/ftuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/ftuser/J7.0
BFG_DATA=/u/ftuser/DATA_DIR
export PATH STEPLIB BFG_JAVA_HOME BFG_DATA
```



Attention: The LIBPATH environment variable is no longer needed when calling **fte*** commands from a z/OS UNIX environment, and should be removed from any existing .wmqfterc script

Optionally, you can also set the following environment variables:

Table 50. Optional z/OS environment variable

Environment variable	Value
BFG_WTO	<p>One of the following values will enable BFG_WTO :</p> <ul style="list-style-type: none"> • YES • ON • TRUE <p>One of the following values will disable BFG_WTO. These values are not case sensitive.</p> <ul style="list-style-type: none"> • NULL • NO • OFF • FALSE <p>Enables z/OS logging. By default, this environment variable is disabled.</p> <p>Messages that are written to the agent event log are also written to the z/OS operator log facility, which allows easier access for automation products when you run an agent from JCL. The routing code is Programmer Information (11) and the descriptor code is Informational (12).</p>

Table 50. Optional z/OS environment variable (continued)

Environment variable	Value
BFG_GROUP_NAME	<p>The mqm file group is typically associated with Managed File Transfer configuration data files and commands. Consequently, all users who are members of the mqm group can access, and make changes to the Managed File Transfer configuration. For more information, see File system permissions for MFT in IBM MQ.</p> <p>For a z/OS system, a file group is a z/OS UNIX filesystem entity, and the mqm file group is not necessarily defined. You can define an alternative, existing z/OS UNIX filesystem group for Managed File Transfer configuration data files by using the BFG_GROUP_NAME environment variable. For example, at the z/OS UNIX shell prompt:</p> <pre data-bbox="862 720 1464 793">export BFG_GROUP_NAME=FTEGB</pre> <p>which defines group FTEGB to be associated with any subsequently created configuration files for the current z/OS UNIX session.</p> <p>You can set BFG_GROUP_NAME to a blank value, or remove it.</p>

Configuration properties files

A summary of the properties that are used in Managed File Transfer.

- [The MFT coordination.properties file](#)
- [The MFT command.properties file](#)
- [The MFT agent.properties file](#)
- [Logger configuration properties file](#)

Configuring MFT for the z/OS Automatic Restart Manager (ARM)

Managed File Transfer is an ARM enabled application.

Before you begin

For more information about enabling ARM, and defining ARM policies for your system, see [Using the z/OS Automatic Restart Manager \(ARM\)](#).

If you want to use the MFT DB Logger ability to automatically restart and reconnect to a Db2 database, ARM is the only supported restart manager available.

About this task

Using ARM, agents and loggers can be configured for restart by setting the agent/logger properties armELEMENTYPE, and armELEMENT. Property armELEMENTYPE defines the type of ARM element and property armELEMENT is the name of the element that ARM is to register:

- You can set the agent ELEMTYPE to SYSBFGAG, and armELEMENT can be set to correspond with the agent name.
- You can set the logger ELEMTYPE to SYSBFGLG, and armELEMENT can be set to correspond with the logger name.

Note: Agents and loggers that are configured for restart by ARM can only be successfully run from a batch job or a started task. Attempts to start the agent or logger from the z/OS UNIX System Services command line directly will fail with an ARM error reason code.

Example

The following example of a restart policy defines agent BFGFT7CAG1 as being dependant on queue manager FT7C:

```
RESTART_ORDER
  LEVEL(3)
  ELEMENT_TYPE(SYSBFGAG,SYSBFGLG)

RESTART_GROUP(GROUP7C)
  ELEMENT(SYSMQMGRFT7C)
  ELEMENT(BFGFT7CAG1)
  RESTART_ATTEMPTS(3,300)
```

Example: Creating JCL for Managed File Transfer agents on z/OS

Use this information to generate some JCL that can be used to create and start an agent on IBM MQ for z/OS.

Copy the sample library

Carry out the following procedure:

1. Make a copy of the library SCSQFCMD (see [“Copy SCSQFCMD to create a JCL library” on page 759](#)) by opening the library.

The majority of the members, those that start with BFGX, BFGY, or BFGZ, are templates that you use to generate the customized JCL for the agent later on.

The important member is BFGCOPY.

2. Open BFGCOPY and replace:

++supplied_library++

with the name of the SCSQFCMD library that was installed as part of the product.

++service-library++

with the name of the library that you want to use for your agent (the target library).

3. Submit the job and you have a new library that you can use.

Edit BFGCUSTM

Carry out the following procedure:

1. Open the new library so that you can edit the BFGCUSTM member (see [“Editing member BFGCUSTM” on page 760](#))
2. Modify all of the parameters in the member that are enclosed within ++ characters, and replace them with the appropriate values. For example, change:

++mq_path++

The path to the z/OS UNIX System Services (z/OS UNIX) Components directory. For example, /mqm/V9R2M0.

Note: There are three instances of this variable to replace.

++bfg_data++

To be the z/OS UNIX directory where your IBM MQ Managed File Transfer for z/OS configuration is to be stored.

++service_type++

To the word AGENT

++agent_name++

To be the name of your agent

Notes:

1. Some of the entries, such as ++options++ required for the CLEAN_AGENT_PROPS, are not needed and so you should remove these.
2. See [“Prima di avviare la configurazione di MFT per z/OS”](#) on page 754 for a complete list of all of the parameters in the BFGCUSTM member, along with a description of what values they should have.

Submit the BFGCUSTM JCL

Carry out the following procedure:

1. Submit the job.
2. Exit the library in ISPF.

This is necessary because the BFGCUSTM job is updating the library, and cannot do that while the library is open.

3. When the job completes look at the joblog.

You will see a number of messages, indicating that new members have been created within the library.

Each of these members contains JCL that can be used to perform specific tasks for your agent. See [“z/OS agent and logger command JCL scripts”](#) on page 768 for a list of these members, along with the IBM MQ Managed File Transfer commands that they correspond to.

Submit BFGAGCR to create the agent

The new member BFGAGCR contains some JCL that [creates an agent](#) by invoking the **fteCreateAgent** command.

Carry out the following procedure:

1. Open up member BFGAGCR.

You should see that BFGAGCR has been populated with the name of your:

- Agent
- Agent queue manager
- The coordination queue manager for the MFT topology

2. Submit member BFGAGCR.

When the member runs, it:

- Creates the required configuration files for your agent.
- Connects to the agent queue manager, and creates the system queues that the agent needs, using CSQUTIL.
- Registers the agent with the coordination queue manager.

Start the agent by submitting BFGAGST

Carry out the following procedure:

1. Submit the BFGAGST member. See [using the agent](#) for various commands that show you that the agent is working correctly.

2. When the job completes, check the joblog contains the following messages:

```
BFGAG0058I: The agent has successfully initialized.  
BFGAG0059I: The agent has been successfully started.
```

which means that your agent is up, running, and ready to perform managed transfers.

Moving an MFT agent to a new z/OS LPAR

It is sometimes necessary to move an IBM MQ Managed File Transfer for z/OS agent from one LPAR to another, while keeping the agent in the same IBM MQ Managed File Transfer topology with the same coordination and command queue managers. The steps needed to do this depend on how the agent being migrated was originally created.

About this task

Move your IBM MQ Managed File Transfer for z/OS agent in one of the following ways:

- If the agent was originally created using a customized version of the SCSQFCMD library, use the library to recreate it on a new LPAR.
- If the agent was originally created by running z/OS UNIX System Services (z/OS UNIX) commands, use the commands to recreate it on a new LPAR.

Note:

Scheduled transfers and transfer templates are stored on the coordination queue manager for an IBM MQ Managed File Transfer topology. This task assumes that the coordination queue manager is not part of the movement work. In this case, any scheduled transfers and transfer templates associated with the agent being moved remain on the existing coordination queue manager after the move is completed.

Procedure

- Move an agent created using a customized version of the SCSQFCMD library.
If the agent was created using a customized version of the SCSQFCMD library, you can use that library to recreate the IBM MQ Managed File Transfer for z/OS environment, and the agent configuration on the new LPAR. To do this, complete the following steps:
 1. Copy the customized version of the library from the original LPAR to the new LPAR.
 2. Edit the BFGCUSTM member in the customized version of the library on the new LPAR, and make sure that the parameter values are still valid.
 3. Run the BFGCUSTM member on the new LPAR, to create all of the JCL needed to configure the environment and create the agent.
 4. Run the BFGCFR member to define the coordination queue manager to be used by the agent on the new LPAR, and create the directory structure needed to store the IBM MQ Managed File Transfer configuration.
 5. Next, run the BFGCMCR member, to define the command queue manager to be used by the agent on the new LPAR.
 6. Run the BFGAGCR member to recreate the agent and its configuration.
 7. Ensure that the system queues used by the agent exist on the queue manager for that agent.

If the agent being moved has resource monitors associated with it, you need to recreate the monitors on the new agent. To do this, complete the following steps:

1. On the original LPAR, run the BFGMCLI member to export the definitions for the resource monitor associated with the original agent to XML files.
2. Copy the XML files containing the resource monitor definitions to the new LPAR.

3. Use the BFGMNCRS member in the SCSQFCMD library on the new LPAR to import the resource monitor definitions stored in the XML files. This results in the monitors being created on the new agent.
- Move an agent created by running commands in z/OS UNIX.

If the agent was originally created by running z/OS UNIX commands, you can use commands to recreate the agent on a new LPAR. To do this, complete the following steps:

1. Run the `fteSetupCoordination` command on the new LPAR, to define the coordination queue manager to be used by the agent, and create the directory structure needed to store the IBM MQ Managed File Transfer configuration.
2. Run the `fteSetupCommands` command to define the command queue manager to be used by the agent on the new LPAR.
3. Run the `fteCreateAgent` command to recreate the agent and its configuration.
4. Ensure that the system queues used by the agent exist on the queue manager for that agent.

If the agent being moved has resource monitors associated with it, you need to recreate the monitors on the new agent. To do this, complete the following steps:

1. On the original LPAR, run the `fteListMonitors` command, specifying the `-ox` parameter, to export the definitions for the resource monitor, associated with the original agent, to XML files.
2. Copy the XML files containing the resource monitor definitions to the new LPAR.
3. Run the `fteCreateMonitor` command on the new LPAR, specifying the `-ix` parameter, to import the resource monitor definitions stored in the XML files. This results in the monitors being created on the new agent.

Planning your MFT infrastructure with IBM MQ for z/OS queue sharing groups

You need to consider the following, if you are using IBM MQ Managed File Transfer (MFT), when one or more of the agents, command or coordination queue managers are part of an IBM MQ for z/OS queue sharing group.

See [MFT topology overview](#) for a description of agents, command queue managers, and coordination queue managers.

Agent queue managers

Normally an MFT agent connects to a single agent queue manager, and uses local queues that are only accessible by that queue manager. The agent is informed which queue manager to connect to, by providing it with the queue manager name when the agent is first created.

With IBM MQ for z/OS, it is possible to create the agent and replace the queue manager name with the name of a queue sharing group (QSG). This means that the agent can connect to any available queue manager in the QSG to perform file transfers. Should there be a failure of the queue manager that the agent is currently connected to, the agent detects the failure and reconnects to an alternative queue manager in the QSG.

Connecting an agent to a QSG in combination with the highly available agent support provided allows very robust MFT topologies to be created. See [“Agent ad alta disponibilità in Managed File Transfer” on page 802](#).

For example, in the following figure *Agent1* has been created so that its agent queue manager is a QSG consisting of two queue managers *QM1* and *QM2*. The agent queues have been defined as shared queues, stored in the coupling facility.

This means that the agent can run on either *LPAR 1* or *LPAR 2* and connect to either *QM1* or *QM2*. The files and data sets that the agent reads from, or writes to, are shared, meaning they can be accessed from either LPAR.

In addition, the agent has been configured to be a highly available agent. In the diagram the agent is active in *LPAR 1* and a standby instance of the agent is running in *LPAR 2*.

This topology provides high resilience. Should either the agent running on *LPAR 1* fail, or queue manager *QM1* fail, or *LPAR 1* fail the standby instance of the agent on *LPAR 2* can take over and carry on processing file transfers from the point of failure.

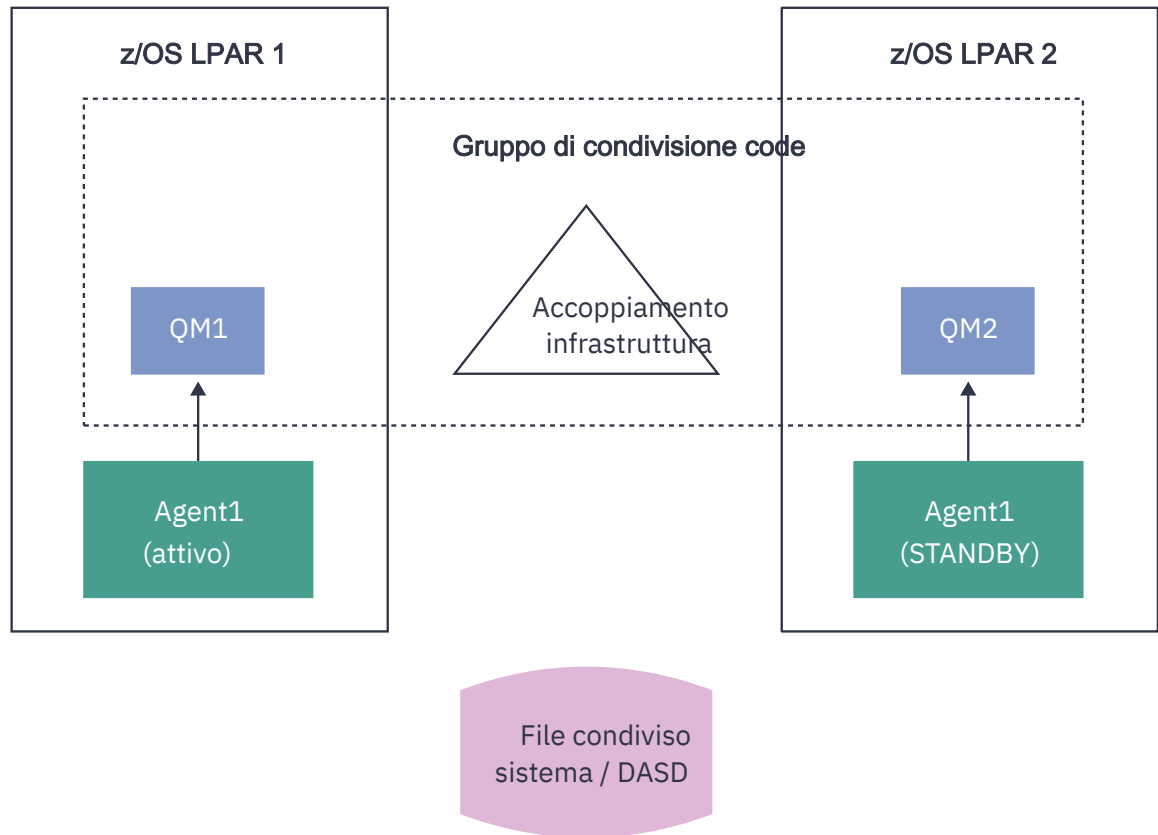


Figure 95. Highly available MFT agent using a queue sharing group

Creating an agent that uses a QSG as the agent queue manager

You create an agent using the `fteCreateAgent` command. When doing this, the name of the queue sharing group is provided for the agent queue manager. For example:

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1
```

This creates an agent called *Agent1* which uses any queue manager that is a member of QSG *QSG1* as its agent queue manager. In this configuration the agent connects to the agent queue manager using a cross memory (bindings mode) connection which means that the agent and the queue manager must be on the same LPAR. This is exactly like the example shown in figure 1 above.

When you run the **fteCreateAgent** command it generates a set of MQSC commands to create the necessary queues on the agent queue manager.

When the agent queue manager is a QSG, this set of commands needs to be modified so that each queue is created as a shared queue. That is, each queue needs to be created with `QSGDISP(SHARED)` and an appropriate coupling facility structure provided by the `CFSTRUCT` attribute.

The following example shows you how to change the MQSC command for creating the `SYSTEM.FTE.COMMAND.AGENT1` queue as a shared queue. The changes to the defaults are in bold text.

Important: You need to make similar changes to all the other queues that the agent uses.

```
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.AGENT1) +
  QSGDISP(SHARED) +
  CFSTRUCT(MFTSTRUCT) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  INDXTYPE(CORRELID) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(99999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
```

Creating an agent that uses a QSG as the agent queue manager and connects as a client

Agents can connect to their agent queue manager using a client channel. You can use this approach to allow the agent to run on distributed platforms while connecting to a QSG. If all queue managers in the QSG are licensed for IBM MQ Advanced for z/OS Value Unit Edition, then the agent can also connect to them from a z/OS LPAR that does not have a local queue manager.

This topology is shown in the following figure and allows the agent to take advantage of the resiliency of QSGs. If the queue manager in the QSG that the agent is currently connected to fails, then the agent automatically reconnects to a different member of the QSG and carries on processing.

The sysplex distributor is used to spread the connections from the agent across the available queue managers in the QSG.

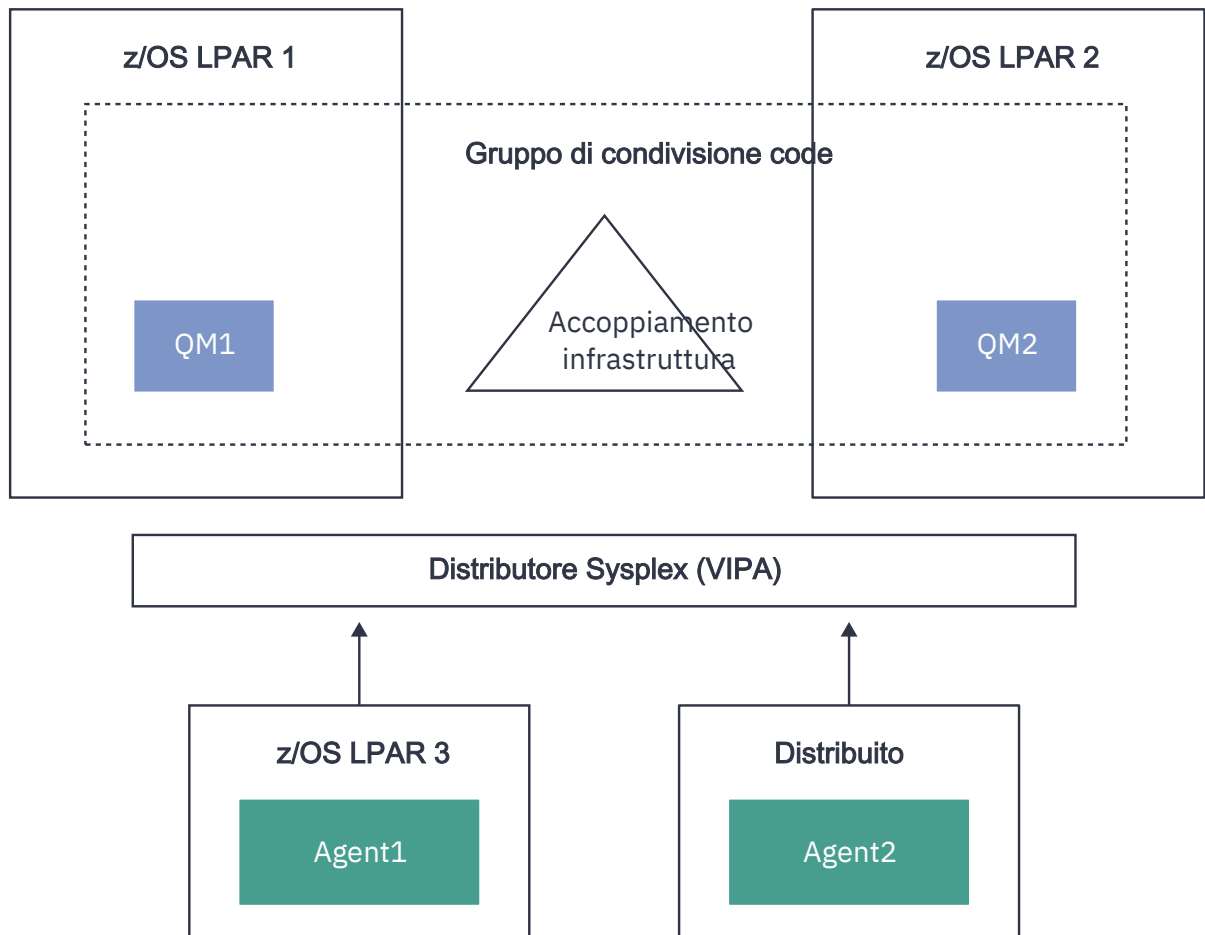


Figure 96. MFT agents connecting to a queue sharing group as a client

In order to make use of this topology, the queue managers in the QSG must each have a server connection channel defined for use by the agent. See [“Connecting a client to a queue sharing group”](#) on page 63 for information on how to do this.

When creating the agent the queue managers need to be configured, so that they can use the channel that is defined to the QSG, and access it through the sysplex distributor. For example:

```
fteCreateAgent -agentName Agent1 -agentQMgr QSG1 -agentQMgrHost vipaAddress
-agentQMgrPort sharedPort -agentQMgrChannel CHANNEL1
```

As previously stated the MQSC commands generated by running the **fteCreateAgent** command must be adjusted to specify QSGDISP(SHARED) and an appropriate coupling facility structure in the CFSTRUCT attribute.

Command queue managers

The MFT command queue manager can be part of a QSG. However, the name of a QSG cannot be used when specifying a command queue manager; you must use a specific queue manager name.

Coordination queue managers

The MFT coordination queue manager can be part of a QSG. However, as with a command queue manager, the name of a QSG cannot be used when specifying a coordination queue manager; you must use a specific queue manager name.

Commands connecting to a QSG

MFT provides a number of commands to managed agents, transfers and agent, command, or coordination queue managers. You can use only those commands that connect to an agent queue manager, if the queue manager is in a QSG.

Following is a list of the commands that connect to the agent queue manager:

- **fteCleanAgent**
- **fteCreateAgent**
- **fteCreateBridgeAgent**
- **fteCreateCDAgent**
- **fteDeleteAgent**

Note that you must provide the name of the queue manager when running other MFT commands.

Using Managed File Transfer for z/OS with the JZOS Java launcher

You can apply the instructions in this topic as an alternative method of using Managed File Transfer in your enterprise, on your IBM MQ for z/OS system.

Overview

Managed File Transfer for z/OS (MFT) uses the standard z/OS installation procedure. An alternative way of running MFT commands is to use JCL and the JZOS Java Launcher.

See [JZOS Batch Launcher and Toolkit](#) for further details.

If your JCL fails to process correctly, see [Common MFT problems with JZOS](#).

Example JCL

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL) (1)
// INCLUDE MEMBER=BFGJCL8 (2)
// DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}" (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

where:

- (1) Is the location of included JCL statements
- (2) Include the specified JCL member from the location in 1)
- (2A) This extends the //STDENV - see below
- (3) This is the command to be executed, without the leading fte prefix
- (4) These lines are required, they set up information for JZOS
- (5) The parameters to the command
- The BFGJCL8 member (you can select your own name) invokes JZOS. This member has the STEPLIB and other JCL needed to run MFT.

Other JCL you need to include

You should include JCL for the IBM MQ for z/OS libraries, and if you are using the Db2 logger, the Db2 libraries.

For example:

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T' (1)
//STÉPLIB DD DSN=SYS1.SIEALNKE,DISP=SHR (2)
//* MQ libraries
// DD DSN=MQM.V920.SCSQAUTH,DISP=SHR MQ Bindings
// DD DSN=MQM.V920.SCSQANLE,DISP=SHR MQ Bindings
// DD DSN=MQM.V920.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
// DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNEXIT.DBCP
// DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD
// DD DISP=SHR,DSN=SYS2.DB2.V12.SDSNLOAD2
//SYSOUT DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT DD SYSOUT=H
//STDERR DD SYSOUT=H

//STDENV DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR (3)
```

where:

- (1) Is the name of the JZOS program. Look in SYS1.SIEALNKE for the version on your system. Add ,PARM='+T' to give additional diagnostics.
- (2) This is the data set with the JZOS program.
- (3) This is the member name of a shell script. It defines parameters needed by MFT. See [“Shell script to define MFT” on page 784](#).

It can be any data set and member. It needs to be last in the file because the JCL job extends this. See 2A in [“Example JCL” on page 783](#).

Shell script to define MFT

In the [“Other JCL you need to include” on page 783](#) example, the member BFGZENV8 is used. This is based on the JZOS profile.

You need to know:

- The location where Java is installed
- The location of the IBM MQ for z/OS Java libraries and the MFT libraries.
- A user ID needs to be in a specific group to be considered as an IBM MQ for z/OS administrator. You need the name of this group
- If you are not using English for the messages, you need to know which language to specify.

Example file

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH:"/mqm/V9R2M0/java/lib/"
export LIBPATH

export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/mqm/V9R2M0/mqft"
export BFG_CONFIG="/u/johndoe/fteconfig"
```



```
# export BFG_LANG=" -Duser.language=de "  
export BFG_LANG=" "
```

where:

export _BPXK_AUTOCVT="ON"

Is required for Unicode conversion

export JAVA_HOME="/java/java71_bit64/J7.1_64/"

Is the location of the Java directory. Specify the name of the path for Java. This directory contains bin and other directories.

export PATH="/bin:\${JAVA_HOME}/bin/classic/"

Sets up the path statement for Java executable statements

LIBPATH="/lib:/usr/lib:\${JAVA_HOME}/bin"

Sets up the library path for the Java executable statements

LIBPATH="\$LIBPATH:\${JAVA_HOME}/bin/classic"

Adds more Java libraries to the LIBPATH statement.

LIBPATH=\$LIBPATH:"/mqm/V9R2M0/java/lib/"

Adds IBM MQ for z/OS libraries in the library path. Specify the name of your IBM MQ for z/OS libraries in z/OS UNIX System Services.

export LIBPATH

Makes the LIBPATH available to JZOS

export BFG_JAVA_HOME="\${JAVA_HOME}"

Sets the BFG_JAVA_HOME to the value of JAVA_HOME specified above

export BFG_WTO="YES"

Setting BFG_WTO to YES causes messages to be displayed on the joblog using WTO

export BFG_GROUP_NAME=MQADM

User IDs, which are a member of the specified group, are considered IBM MQ for z/OS administrators

export BFG_PROD="/mqm/V9R2M0/mqft"

Is the path where the MFT code is located

export BFG_DATA="/u/johndoe/fteconfig"

Is where the MFT configuration information is stored

export BFG_LANG=" -Duser.language=de "

Is a commented out statement to define the language as German

export BFG_LANG=" "

Specifies the language as the default, English.

The contents of the MFT product in `/lib/messages/BFGNVMessages_*.properties` lists the languages available. The default is to leave the value blank, which means that English is used.

Related tasks

[“Configuring Managed File Transfer for z/OS” on page 753](#)

Managed File Transfer for z/OS requires customization to enable the component to operate correctly.

[Planning for Managed File Transfer](#)

IBM i Configurazione di MFT su IBM i

Per iniziare a utilizzare Managed File Transfer dopo averlo installato, è necessario completare una configurazione per il gestore code di coordinamento e l'agent.

Informazioni su questa attività

Dopo l'installazione, è necessario eseguire gli script di configurazione forniti da Managed File Transfer per i nuovi gestori code di coordinamento e i nuovi agenti prima di poter utilizzare i gestori code di coordinamento e gli agent per trasferire i file. È quindi necessario avviare gli agent creati.

Procedura

1. Per tutti i nuovi gestori code di coordinamento: eseguire i comandi MQSC nel file `coordination_qmgr_name.mqsc` rispetto al gestore code di coordinamento. Se il gestore code di coordinamento non si trova sullo stesso computer dell'installazione, copiare il file di script MQSC sul computer su cui si trova il gestore code, quindi eseguire lo script.

- a) Da una riga comandi IBM i , avviare qshell utilizzando il seguente comando: `CALL QSHELL`
- b) Modificare la seguente directory: `/QIBM/UserData/mqm/mqft/config/coordination_qmgr_name`
- c) Immettere il seguente comando, sostituendo `coordination_qmgr_name` con il nome del proprio gestore code:

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
```

È possibile invece configurare manualmente il gestore code di coordinamento. Per ulteriori informazioni, consultare [“Configurazione del gestore code di coordinamento per MFT”](#) a pagina 790.

2. Per tutti i nuovi agent: eseguire i comandi MQSC nel file `agent_name_create.mqsc` sul gestore code dell'agent.

Se il gestore code dell'agent non si trova sullo stesso computer dell'agent, copiare il file di script MQSC sul computer in cui si trova il gestore code ed eseguire lo script.

- a) Da una riga comandi IBM i , avviare qshell utilizzando il seguente comando: `CALL QSHELL`
- b) Modificare la seguente directory: `/QIBM/UserData/mqm/mqft/config/agent_qmgr_name/agents`
- c) Immettere il seguente comando, sostituendo `agent_qmgr_name` con il nome del gestore code dell'agent e sostituendo `agent_name` con il nome dell'agent:

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
```

È invece possibile configurare manualmente il gestore code agent. Per ulteriori informazioni, fare riferimento a [“Configurazione dei gestori code dell'agent MFT”](#) a pagina 797.

3. Se il sottosistema QMFT non è stato ancora avviato come parte dell'installazione, dalla riga comandi IBM i , avviare il sottosistema QMFT utilizzando il seguente comando: `STRSBS SBS(DQM/MMFT/QMFT)` o `STRSBS QM/MMFT/QMFT`
4. Avviare i nuovi agent utilizzando il comando **`fteStartAgent`** .
 - a) Da una riga comandi IBM i , avviare qshell utilizzando il seguente comando: `CALL QSHELL`
 - b) Modificare la seguente directory: `/QIBM/ProdData/mqm/bin`
 - c) Immettere il seguente comando, sostituendo AGENT con il nome del proprio agente:

```
./fteStartAgent AGENT
```

Operazioni successive

Si consiglia di impostare le sandbox per limitare le aree del file system a cui un agent può accedere. Questa funzione è descritta in [Utilizzo delle sandbox dell'agent MFT](#).

Concetti correlati

[“Configurazione di MFT per il primo utilizzo”](#) a pagina 786

È necessario eseguire alcune attività di configurazione per gli agent Managed File Transfer e i gestori code una volta, la prima volta che si desidera utilizzarli.

Configurazione di MFT per il primo utilizzo

È necessario eseguire alcune attività di configurazione per gli agent Managed File Transfer e i gestori code una volta, la prima volta che si desidera utilizzarli.

Concetti correlati

[“Connessione a IBM MQ” a pagina 787](#)

Tutte le comunicazioni di rete con i gestori code IBM MQ , incluse le comunicazioni correlate a Managed File Transfer, riguardano i canali IBM MQ . Un canale IBM MQ rappresenta un'estremità di un collegamento di rete. I canali sono classificati come canali di messaggi o canali MQI.

[“Configurazione di un gestore code a più istanze per utilizzare MFT” a pagina 793](#)

IBM WebSphere MQ 7.0.1 in poi supporta la creazione di gestori code a più istanze. Un gestore code a più istanze si riavvia automaticamente su un server standby. Managed File Transfer supporta la connessione ai gestori code dell'agent a più istanze, a un gestore code di coordinamento a più istanza e a un gestore code di comando a più istanze.

Attività correlate

[“Configurazione dei gestori code di rete MFT” a pagina 788](#)

Se la propria rete Managed File Transfer include più di un gestore code IBM MQ , tali gestori code IBM MQ devono essere in grado di comunicare in remoto tra loro.

[“Configurazione dei gestori code dell'agent MFT” a pagina 797](#)

In seguito all'installazione, eseguire lo script *agent_name_create.mqsc* nella directory *MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name* per eseguire la configurazione necessaria per il gestore code dell'agent. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code dell'agent.

[“Configurazione del gestore code di coordinamento per MFT” a pagina 790](#)

Dopo aver eseguito il comando **fteSetupCoordination** , eseguire lo script *coordination_qmgr_name.mqsc* nella directory *MQ_DATA_PATH/mqft/config/coordination_qmgr_name* per eseguire la configurazione necessaria per il gestore code di coordinamento. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code di coordinamento.

[“Creating an MFT Agent or Logger command data set” a pagina 752](#)

You can create a PDSE data set of commands from the Managed File Transfer command template data set for a specific Managed File Transfer Agent or Managed File Transfer Logger for a specific coordination.

[“Updating an existing MFT Agent or Logger command data set on z/OS” a pagina 764](#)

You can update an Managed File Transfer command PDSE library data set that is created from the Managed File Transfer command template data set.

Riferimenti correlati

[MFT Impostazioni coda agent](#)

[Code di sistema MFT e argomento di sistema](#)

[“Conservazione dei messaggi di log MFT” a pagina 795](#)

Managed File Transfer invia l'avanzamento del trasferimento file e le informazioni di log al gestore code di coordinamento. Il gestore code di coordinamento pubblica queste informazioni in tutte le sottoscrizioni corrispondenti al SISTEMA.SYSTEM.FTE . Se non ci sono sottoscrizioni, queste informazioni non vengono conservate.

Connessione a IBM MQ

Tutte le comunicazioni di rete con i gestori code IBM MQ , incluse le comunicazioni correlate a Managed File Transfer, riguardano i canali IBM MQ . Un canale IBM MQ rappresenta un'estremità di un collegamento di rete. I canali sono classificati come canali di messaggi o canali MQI.

Managed File Transfer e canali

Managed File Transfer utilizza canali MQI per connettere gli agent in modalità client ai relativi gestori code agent e per connettere le applicazioni di comando (ad esempio, **fteCreateTransfer**) ai relativi gestori code di comando e coordinamento. Nella configurazione predefinita, queste connessioni vengono effettuate utilizzando un canale SVRCONN denominato SYSTEM.DEF.SVRCONN, che esiste per impostazione predefinita su tutti i gestori code. A causa di questi valori predefiniti, non è necessario modificare i canali MQI per un'installazione di Managed File Transfer di base.

Esistono sei tipi di endpoint del canale di messaggi, ma questo argomento riguarda solo le coppie mittente - destinatario. Consultare [Componenti di accodamento distribuiti](#) per informazioni su altre combinazioni di canali.

Percorsi dei messaggi richiesti

I messaggi IBM MQ possono viaggiare solo su canali di messaggi, quindi è necessario assicurarsi che i canali siano disponibili per tutti i percorsi di messaggi richiesti da Managed File Transfer. Questi percorsi non devono essere diretti; i messaggi possono viaggiare attraverso i gestori code intermedi, se necessario. Questo argomento riguarda solo la comunicazione diretta punto a punto. Per ulteriori informazioni su queste opzioni, consultare [Come accedere al gestore code remoto](#).

I percorsi di comunicazione utilizzati da Managed File Transfer sono i seguenti:

Da agent a agent

I due agente tra cui vengono trasferiti i file richiedono la comunicazione bidirezionale tra i gestori code associati. Poiché questo percorso trasporta i dati di massa, considerare di rendere il percorso il più breve, veloce o economico possibile in base alle proprie necessità.

Da agente a coordinamento

I messaggi di log dagli agenti che partecipano a un trasferimento devono essere in grado di raggiungere il gestore code di coordinamento.

Comando per l'agente

Qualsiasi gestore code a cui si collegano le applicazioni di comandi o IBM MQ Explorer (utilizzando il gestore code comandi) deve essere in grado di inviare messaggi ai gestori code degli agent che tali applicazioni di comandi vengono utilizzate per il controllo. Per consentire ai messaggi di feedback di essere visualizzati dai comandi, utilizzare una connessione bidirezionale.

Per ulteriori informazioni, consultare [Verifica di un'installazione di IBM MQ](#) per la piattaforma o le piattaforme utilizzate dall'azienda.

Concetti correlati

[“Configurazione di un gestore code a più istanze per utilizzare MFT” a pagina 793](#)

IBM WebSphere MQ 7.0.1 in poi supporta la creazione di gestori code a più istanze. Un gestore code a più istanze si riavvia automaticamente su un server standby. Managed File Transfer supporta la connessione ai gestori code dell'agent a più istanze, a un gestore code di coordinamento a più istanza e a un gestore code di comando a più istanze.

Attività correlate

[“Configurazione dei gestori code di rete MFT” a pagina 788](#)

Se la propria rete Managed File Transfer include più di un gestore code IBM MQ, tali gestori code IBM MQ devono essere in grado di comunicare in remoto tra loro.

[“Configurazione del gestore code di coordinamento per MFT” a pagina 790](#)

Dopo aver eseguito il comando **fteSetupCoordination**, eseguire lo script `coordination_qmgr_name.mqsc` nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` per eseguire la configurazione necessaria per il gestore code di coordinamento. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code di coordinamento.

Configurazione dei gestori code di rete MFT

Se la propria rete Managed File Transfer include più di un gestore code IBM MQ, tali gestori code IBM MQ devono essere in grado di comunicare in remoto tra loro.

Informazioni su questa attività

Esistono due modi per configurare i gestori code in modo che siano in grado di comunicare tra loro:

- Impostando un cluster di gestori code IBM MQ.

Per informazioni sui cluster del gestore code IBM MQ e su come configurarli, fare riferimento a [“Configurazione di un cluster di gestore code” a pagina 303](#).

- Impostando i canali tra i gestori code, descritto di seguito:

Impostazione dei canali tra i gestori code

Configurare i seguenti canali di messaggi tra gestori code:

- Dal gestore code dell'agente al gestore code di coordinazione
- Dal gestore code comandi al gestore code agent.
- Dal gestore code dell'agente al gestore code comandi (per consentire la visualizzazione dei messaggi di feedback da parte dei comandi).
- Dal gestore code comandi al gestore code di coordinamento
- Dal gestore code dell'agent a qualsiasi altro gestore code dell'agent nella rete Managed File Transfer

Se sono necessarie ulteriori informazioni su come impostare questa comunicazione, iniziare con queste informazioni: [Amministrazione di oggetti IBM MQ remoti mediante MQSC](#).

Alcuni passi di esempio consigliati sono:

Procedura

1. Creare una coda di trasmissione sul gestore code IBM MQ con lo stesso nome del gestore code di coordinamento.

È possibile utilizzare il seguente comando MQSC:

```
DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
```

2. Sul gestore code IBM MQ , creare un canale mittente per il gestore code di coordinamento Managed File Transfer .

Il nome della coda di trasmissione creata nel passo precedente è un parametro obbligatorio per questo canale.

Per gli agent su Managed File Transfer per IBM MQ, i messaggi vengono pubblicati con un formato vuoto.

È possibile utilizzare il seguente comando MQSC:

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME(' coordination-qmgr-host(coordination-qmgr-port) ' )  
XMITQ(coordination-qmgr-name) CONVERT(NO)
```

Nota: Impostare CONVERT (NO), solo se richiesto.

3. Sul gestore code di coordinamento Managed File Transfer , creare un canale ricevente per il gestore code IBM MQ . Assegnare a questo canale ricevente lo stesso nome del canale mittente sul gestore code IBM MQ .

È possibile utilizzare il seguente comando MQSC:

```
DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
```

Operazioni successive

Successivamente, attenersi alla procedura di configurazione per il gestore code di coordinamento: [“Configurazione del gestore code di coordinamento per MFT” a pagina 790](#).

Concetti correlati

[“Connessione a IBM MQ” a pagina 787](#)

Tutte le comunicazioni di rete con i gestori code IBM MQ , incluse le comunicazioni correlate a Managed File Transfer, riguardano i canali IBM MQ . Un canale IBM MQ rappresenta un'estremità di un collegamento di rete. I canali sono classificati come canali di messaggi o canali MQI.

[“Configurazione di un gestore code a più istanze per utilizzare MFT” a pagina 793](#)

IBM WebSphere MQ 7.0.1 in poi supporta la creazione di gestori code a più istanze. Un gestore code a più istanze si riavvia automaticamente su un server standby. Managed File Transfer supporta la connessione ai gestori code dell'agent a più istanze, a un gestore code di coordinamento a più istanza e a un gestore code di comando a più istanze.

Attività correlate

[“Configurazione del gestore code di coordinamento per MFT” a pagina 790](#)

Dopo aver eseguito il comando **fteSetupCoordination**, eseguire lo script `coordination_qmgr_name.mqsc` nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` per eseguire la configurazione necessaria per il gestore code di coordinamento. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code di coordinamento.

Configurazione del gestore code di coordinamento per MFT

Dopo aver eseguito il comando **fteSetupCoordination**, eseguire lo script `coordination_qmgr_name.mqsc` nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` per eseguire la configurazione necessaria per il gestore code di coordinamento. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code di coordinamento.

Informazioni su questa attività

Procedura

1. Creare una coda locale denominata SYSTEM.FTE.
2. Aggiungere SYSTEM.FTE nella coda SYSTEM.QPUBSUB.QUEUE.NAMELIST.
3. Creare un argomento denominato SYSTEM.FTE con una stringa di argomenti SYSTEM.FTE.
4. Verificare gli attributi Consegna messaggi non persistenti (NPMMSGDLV) e Consegna messaggi persistenti (PMSGDLV) di SYSTEM.FTE. FTE è impostato su ALLAVAIL.
5. Verificare che l'attributo della modalità di pubblicazione / sottoscrizione (PSMODE) del gestore code di coordinamento sia impostato su ENABLED.

Operazioni successive

Se si esegue il comando `strmqm -c` su un gestore code configurato come gestore code di coordinamento, il comando elimina la modifica apportata al [passo 2](#) (aggiungendo il SISTEMA SYSTEM.FTE nella coda SYSTEM.QPUBSUB.QUEUE.NAMELIST elenco nomi). Ciò è dovuto al fatto che `strmqm -c` ricrea gli oggetti IBM MQ predefiniti e inverte le Managed File Transfer modifiche. Pertanto, se il gestore code è stato avviato con `strmqm -c`, completare una delle seguenti operazioni:

- Eseguire nuovamente lo script `coordination_qmgr_name.mqsc` sul gestore code.
- Ripetere il [passo 2](#).

Concetti correlati

[“Connessione a IBM MQ” a pagina 787](#)

Tutte le comunicazioni di rete con i gestori code IBM MQ, incluse le comunicazioni correlate a Managed File Transfer, riguardano i canali IBM MQ. Un canale IBM MQ rappresenta un'estremità di un collegamento di rete. I canali sono classificati come canali di messaggi o canali MQI.

[“Configurazione di un gestore code a più istanze per utilizzare MFT” a pagina 793](#)

IBM WebSphere MQ 7.0.1 in poi supporta la creazione di gestori code a più istanze. Un gestore code a più istanze si riavvia automaticamente su un server standby. Managed File Transfer supporta la connessione ai gestori code dell'agent a più istanze, a un gestore code di coordinamento a più istanza e a un gestore code di comando a più istanze.

Attività correlate

[“Configurazione dei gestori code di rete MFT” a pagina 788](#)

Se la propria rete Managed File Transfer include più di un gestore code IBM MQ , tali gestori code IBM MQ devono essere in grado di comunicare in remoto tra loro.

Riferimenti correlati

[Coordinamento fteSetup](#)

Creazione di una IBM MQ File Transfer Structure

È possibile configurare una struttura Managed File Transfer , basata su un singolo agent connesso a un gestore code sulla stessa macchina.

Informazioni su questa attività

La configurazione di MFT viene memorizzata in una struttura file in IBM MQ DataPath, sulla macchina su cui si trova l'agent.

La seguente configurazione di esempio è per un gestore code MFT in IBM MQ 8.0 denominato SAMPLECOORD (con la sicurezza disabilitata) e un singolo agent MFT denominato SAMPLEAGENT:

```
+--- config
  +--- SAMPLECOORD
        +--- command.properties
        +--- coordination.properties
        +--- SAMPLECOORD.mqsc
        +--- agents
            +--- SAMPLEAGENT
                  +--- agent.properties
                  +--- SAMPLEAGENT_create.mqsc
                  +--- SAMPLEAGENT_delete.mqsc

+---logs
  +--- SAMPLECOORD
        +--- agents
            +--- SAMPLEAGENT
                  +--- logs
```

Questo esempio presuppone che la sicurezza del gestore code sia stata disabilitata. I seguenti comandi, eseguiti in **runmqsc**, disabilitano la sicurezza dopo il riavvio del gestore code:

```
runmqsc queue manager
alter qmgr CONNAUTH(NONE);
alter qmgr CHLAUTH(DISABLED);
end;
```

Per la configurazione con la sicurezza abilitata in MFT in IBM MQ 8.0 o versioni successive, **CONNAUTH** richiede tutti i comandi MFT che si connettono a un gestore code per fornire le credenziali ID utente e password. È possibile applicare i parametri aggiuntivi **-mquserid** e **-mqpassword** per ogni comando oppure definire un file `MQMFTCredentials.xml` . Il seguente file di credenziali di esempio definisce l'ID utente di `fteuser`, per cui la password di `MyPassword` deve essere utilizzata durante la connessione al gestore code `SAMPLECOORD`:

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">
  <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPELCOORD"/>
</tns:mqmftCredentials>
```

Per ulteriori informazioni, vedi [Autenticazione della connessioneMFT e IBM MQ](#).

Note:

- Per individuare la propria directory di configurazione MFT , utilizzare il comando **fteDisplayVersion -v** .

- **z/OS** Per gli utenti z/OS , il file `MQMFTCcredential.xml` può essere individuato come membro in un dataset partizionato con formato record variabile (RECFM = V) o formato record non definito (RECFM = U).
- Per la configurazione con la sicurezza abilitata, aggiungere il seguente parametro ai seguenti passi per associare le credenziali con il gestore code pertinente: `-F full_credential_file_path`.
- La password in testo chiaro in `MQMFTCcredential.xml` può essere offuscata utilizzando il seguente comando:

```
fteObfuscate -f full_file_path_to_MQMFTCcredentials.xml
```

Procedura

1. Creare un gestore code di coordinamento.

Un gestore code di coordinamento è un singolo gestore code, utilizzato per ricevere tutte i log di trasferimento e le informazioni sullo stato dagli agent. Esegui il seguente comando:

```
fteSetupCoordination -coordinationQMGr coordination_qmgr_name
```

Questo crea la configurazione di base di livello superiore e crea un IBM MQ file script da richiamare `coordination_qmgr_name.mqsc`.

La configurazione deve quindi essere caricata nel gestore code, eseguendo il seguente comando IBM MQ :

```
runmqsc queue_manager_name < coordination_qmgr_name.mqsc
```

Nota: Per la connessione del client TCP a un gestore code, è possibile utilizzare:

```
fteSetupCoordination -coordinationQMGr coordination_qmgr_name
-coordinationQMGrHost coordination_qmgr_host -coordinationQMGrPort coordination_qmgr_port
-coordinationQMGrChannel coordination_qmgr_channel
```

Per `coordination_qmgr_name.mqsc` creato, sarà necessario eseguire il comando **runmqsc** sulla stessa macchina su cui è in esecuzione il gestore code di coordinamento.

2. Creare il gestore code comandi.

Un gestore code comandi è un singolo gestore code che è stato preconfigurato in modo che l'infrastruttura IBM MQ possa instradare le richieste MFT all'agent pertinente. Esegui il seguente comando:

```
fteSetupCommands -connectionQMGr Command QM Name -p Coordination QM Name
```

Questo crea un file `command.properties` nella directory di coordinazione. Si noti che il `-p` è facoltativo e non è richiesto se i comandi vengono impostati per il coordinamento predefinito.

Nota: Per la connessione del client TCP a un gestore code, è possibile utilizzare:

```
fteSetupCommands -p coordination_qmgr_name -commandQMGr connection_qmgr_name
-commandQMGrHost connection_qmgr_host -commandQMGrPort connection_qmgr_port
-commandQMGrChannel connection_qmgr_channel
```

3. Creare l'agent.

Un agent è un'applicazione che può inviare e ricevere file. Esegui il seguente comando:

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMGr agent_qmgr_name
```


Ciò crea la configurazione dell'agent sotto il coordinamento e crea un file script IBM MQ per richiamare `agent_name.mqsc` nella directory di configurazione dell'agent.

Eseguire questo comando IBM MQ per caricare il file script IBM MQ nel gestore code:

```
runmqsc agent_qmgr_name < agent_name_create.mqsc file
```

Nota: Per la connessione del client TCP a un gestore code, è possibile utilizzare:

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMGr agent_qmgr_name  
-agentQMGrHost agent_qmgr_host -agentQMGrPort agent_qmgr_port -agentQMGrChannel  
agent_qmgr_channel
```

4. Avviare l'agent.

Esegui il seguente comando:

```
fteStartAgent -p coordination_qmgr_name agentName
```

L'agent viene avviato in background e viene restituito il prompt dei comandi. Per verificare che l'agente sia in esecuzione, eseguire il seguente comando:

```
ftelistAgents -p coordination_qmgr_name
```

Mostra lo stato degli agent. Se l'esecuzione dell'agent ha esito positivo, viene riportato come nello stato READY.

Risultati

Un'infrastruttura MFT di base è pronta per l'uso e ora puoi utilizzare il comando **fteCreateTransfer** per richiedere un trasferimento. In alternativa, se IBM MQ Explorer è disponibile, utilizzare i plug-in MFT per creare e monitorare i trasferimenti.

È possibile aggiungere ulteriori agenti alla configurazione ripetendo il passo 3: Creare l'agente. Se viene utilizzata la connessione client TCP, queste possono trovarsi su macchine differenti. Per macchine differenti, i comandi **fteSetupCoordination** e **fteSetupCommands** devono essere ripetuti per ciascuna macchina, tuttavia non è necessario eseguire gli script mqsc.

Configurazioni più complesse possono avere gestori code separati per il coordinamento e ciascun agent. In questi casi, i vari gestori code dovranno essere connessi tra loro.

Concetti correlati

Operazioni da eseguire se l'agent MFT non è elencato dal comando **ftelistAgents**

Riferimenti correlati

[Coordinamento fteSetup](#)

[Comandi fteSetup: creare il file MFT command.properties](#)

[Agent fteCreate](#)

fteObfuscate: [crittografare i dati sensibili](#)

[Formato file credenziali MFT](#)

[Il file MFTagent.properties](#)

Configurazione di un gestore code a più istanze per utilizzare MFT

IBM WebSphere MQ 7.0.1 in poi supporta la creazione di gestori code a più istanze. Un gestore code a più istanze si riavvia automaticamente su un server standby. Managed File Transfer supporta la connessione ai gestori code dell'agent a più istanze, a un gestore code di coordinamento a più istanza e a un gestore code di comando a più istanze.

Configurazione di un gestore code a più istanze

Importante: Per informazioni sulla configurazione di un gestore code a più istanze IBM MQ , consultare [“Gestori code a più istanze” a pagina 524](#). Accertarsi di aver letto queste informazioni prima di provare a configurare un gestore code a più istanze per l'utilizzo con Managed File Transfer.

Utilizzo di un gestore code a più istanze come gestore code agent

Per consentire a un agente di connettersi sia all'istanza attiva che a quella in standby del gestore code a più istanze, aggiungere la proprietà `agentQMGrStandby` al file `agent.properties` dell'agente. La proprietà `agentQMGrStandby` definisce il nome host e il numero di porta utilizzati per connessioni client per l'istanza del gestore code in standby. Il valore della proprietà deve essere fornito nel formato CONNAME di MQ , ossia `host_name(port_number)`.

La proprietà `agentQMGr` specifica il nome del gestore code a più istanze. La proprietà `agentQMGrHost` specifica il nome host per l'istanza del gestore code attivo e la proprietà `agentQMGrPort` specifica il numero di porta per l'istanza del gestore code attivo. L'agent deve connettersi in modalità client sia all'istanza attiva che all'istanza standby del gestore code a più istanze.

Per ulteriori informazioni, consultare [Il file MFT `agent.properties`](#) .

Questo esempio mostra il contenuto del file `agent.properties` per AGENT1 che si connette a un gestore code a più istanze denominato QM_JUPITER. L'istanza attiva di QM_JUPITER si trova nel sistema host1 e utilizza il numero di porta 1414 per connessioni client. L'istanza standby di QM_JUPITER si trova sul sistema host2 e utilizza il numero porta 1414 per le connessioni client.

```
agentName=AGENT1
agentDesc=
agentQMGr=QM_JUPITER
agentQMGrPort=1414
agentQMGrHost=host1
agentQMGrChannel=SYSTEM.DEF.SVRCONN
agentQMGrStandby=host2(1414)
```

Utilizzo di un gestore code a più istanze come gestore code di coordinamento

Per abilitare le connessioni sia all'istanza attiva che a quella in standby del gestore code di coordinamento a più istanze, aggiungere la proprietà `coordinationQMGrStandby` a tutti i `coordination.properties` file nella topologia Managed File Transfer .

Per ulteriori informazioni, consultare [Il file MFT `coordination.properties`](#) .

Questo esempio mostra il contenuto di un file `coordination.properties` che specifica i dettagli di connessione a un gestore code di coordinamento a più istanze denominato QM_SATURN. L'istanza attiva di QM_SATURN è sul sistema `coordination_host1` e utilizza il numero di porta 1420 per le connessioni client. L'istanza in standby di QM_SATURN si trova sul sistema `coordination_host2` e utilizza il numero di porta 1420 per connessioni client.

```
coordinationQMGr=QM_SATURN
coordinationQMGrHost=coordination_host1
coordinationQMGrPort=1420
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
coordinationQMGrStandby=coordination_host2(1420)
```

Il programma di registrazione autonomo Managed File Transfer deve sempre connettersi al proprio gestore code in modalità bind. Quando si utilizza il programma di registrazione autonomo con un gestore code di coordinamento a più istanze, connettere il programma di registrazione autonomo, in modalità bind, a un gestore code differente. I passi per eseguire questa operazione sono descritti in [“Configurazioni alternative per un logger autonomo MFT” a pagina 820](#). È necessario definire i canali tra il gestore code del programma di registrazione autonomo e il gestore code di coordinamento con il nome host e il numero di porta di entrambe le istanze del gestore code di coordinamento a più istanze. Per informazioni su come svolgere questa procedura, consultare [“Gestori code a più istanze” a pagina 524](#).

Il plug-in Managed File Transfer per IBM MQ Explorer si connette al gestore code di coordinamento in modalità client. Se l'istanza attiva del gestore code di coordinamento a più istanze ha esito negativo, l'istanza in standby del gestore code di coordinamento diventa attiva e il plugin si riconnette.

I Managed File Transfer comandi **fteList*** e **fteShowAgentDetails** si connettono direttamente al gestore code di coordinamento. Se l'istanza attiva del coordinamento a più istanze non è disponibile, questi comandi tenteranno di connettersi all'istanza in standby del gestore code di coordinamento.

Utilizzo di un gestore code a più istanze come gestore code comandi

Per abilitare le connessioni sia all'istanza attiva che a quella in standby del gestore code comandi a più istanze, aggiungere la proprietà `connectionQMgrStandby` a tutti i file `command.properties` nella topologia Managed File Transfer .

Per ulteriori informazioni, consultare [Il file MFT command.properties](#) .

Questo esempio mostra i contenuti di un file `command.properties` che specifica i dettagli di connessione a un gestore code di comandi a più istanze denominato QM_MARS. L'istanza attiva di QM_MARS si trova sul sistema `command_host1` e utilizza il numero di porta 1424 per le connessioni client. L'istanza di standby di QM_MARS si trova sul sistema `command_host2` e utilizza il numero porta 1424 per le connessioni client.

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

Concetti correlati

[“Connessione a IBM MQ” a pagina 787](#)

Tutte le comunicazioni di rete con i gestori code IBM MQ , incluse le comunicazioni correlate a Managed File Transfer, riguardano i canali IBM MQ . Un canale IBM MQ rappresenta un'estremità di un collegamento di rete. I canali sono classificati come canali di messaggi o canali MQI.

Attività correlate

[“Configurazione dei gestori code di rete MFT” a pagina 788](#)

Se la propria rete Managed File Transfer include più di un gestore code IBM MQ , tali gestori code IBM MQ devono essere in grado di comunicare in remoto tra loro.

[“Configurazione del gestore code di coordinamento per MFT” a pagina 790](#)

Dopo aver eseguito il comando **fteSetupCoordination** , eseguire lo script `coordination_qmgr_name.mqsc` nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` per eseguire la configurazione necessaria per il gestore code di coordinamento. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code di coordinamento.

Conservazione dei messaggi di log MFT

Managed File Transfer invia l'avanzamento del trasferimento file e le informazioni di log al gestore code di coordinamento. Il gestore code di coordinamento pubblica queste informazioni in tutte le sottoscrizioni corrispondenti al SISTEMA SYSTEM.FTE . Se non ci sono sottoscrizioni, queste informazioni non vengono conservate.

Modalità per garantire la conservazione delle informazioni

Se l'avanzamento del trasferimento o le informazioni di log sono importanti per la propria azienda, è necessario effettuare una delle seguenti operazioni per garantire che le informazioni vengano conservate:

- Utilizzare il programma di registrazione database Managed File Transfer per copiare i messaggi pubblicati nel SYSTEM.FTE/Log in un database Oracle o Db2 .

- Definire una sottoscrizione al SISTEMA.SYSTEM.FTE , che memorizza le pubblicazioni su una coda IBM MQ . Definire questa sottoscrizione prima di trasferire qualsiasi trasferimento file per garantire che tutti i messaggi di avanzamento e di log vengano conservati nella coda.
- Scrivere un'applicazione che utilizza l'interfaccia MQI (Message Queue Interface) o IBM MQ JMS per creare una sottoscrizione durevole ed elaborare le pubblicazioni consegnate alla sottoscrizione. Questa applicazione deve essere in funzione prima che i file vengano trasferiti per garantire che l'applicazione riceva tutti i messaggi di avanzamento e di log.

Ciascuno di questi approcci è descritto più dettagliatamente nelle seguenti sezioni.

Non basarsi sul plugin IBM MQ Explorer per conservare le informazioni di log.

Utilizzo del programma di registrazione database Managed File Transfer per conservare i messaggi di registrazione

Il programma di registrazione database è un componente facoltativo di Managed File Transfer che è possibile utilizzare per copiare le informazioni di log in un database per scopi di analisi e controllo. Il programma di registrazione database è un'applicazione Java autonoma che viene installata su un sistema che ospita il gestore code di coordinamento e il database. Per ulteriori informazioni sul programma di registrazione database, consultare [“Configurazione di un programma di registrazione MFT”](#) a pagina 807.

Conservazione dell'avanzamento e dei messaggi di log utilizzando il plugin IBM MQ Explorer

Quando un'istanza del plug-in di IBM MQ Explorer viene avviata per la prima volta, l'istanza crea una sottoscrizione durevole sul gestore code di coordinamento. Questa sottoscrizione durevole è utilizzata per raccogliere le informazioni visualizzate nelle viste **Log trasferimenti** e **Avanzamento trasferimento corrente** .

Il nome della sottoscrizione durevole ha il prefisso per mostrare che la sottoscrizione è stata creata dal plugin IBM MQ Explorer MFT , il nome host e il nome utente. Ad esempio MQExplorer_MFT_Plugin_HOST_TJWatson.

Questo prefisso viene aggiunto nel caso in cui un amministratore desideri eliminare una sottoscrizione duratura che non è più in uso attivo da parte di un'istanza del plugin IBM MQ Explorer .

L'uso di una sottoscrizione durevole sul gestore code di coordinamento può causare la creazione di messaggi sul SISTEMA.SYSTEM.MANAGED.DURABLE code. Se si dispone di una rete Managed File Transfer di volumi elevati, utilizzare il plug-in IBM MQ Explorer raramente o entrambi, questi dati del messaggio possono riempire il filesystem locale.

Per evitare che ciò si verifichi, specificare che il plug-in IBM MQ Explorer utilizza una sottoscrizione non durevole al gestore code di coordinamento. Eseguire questa procedura in IBM MQ Explorer:

1. Selezionare **Finestra > Preferenze > MQ Explorer > Managed File Transfer**
2. Dall'elenco **Tipo di sottoscrizione log trasferimenti**, scegliere NON_DURABLE.

Memorizzazione delle pubblicazioni su una coda IBM MQ

Per archiviare i messaggi di log o di avanzamento su una coda IBM MQ , configurare una sottoscrizione sul gestore code di coordinamento che inoltra i messaggi a questa coda. Ad esempio, per inoltrare tutti i messaggi di log a una coda denominata LOG.QUEUE, inoltrare il seguente comando MQSC:

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE)WSHEMA(TOPIC)
```

Dopo che i messaggi di log sono stati inoltrati a una coda IBM MQ , vengono conservati nella coda fino a quando non vengono elaborati da un'applicazione IBM MQ che utilizza la coda.

Scrittura di applicazioni che gestiscono una sottoscrizione duratura al SISTEMA SYSTEM.FTE FTE

È possibile scrivere le applicazioni che gestiscono le proprie sottoscrizioni durevoli in SYSTEM.FTE utilizzando una delle API (application programming interface) supportate da IBM MQ. Queste applicazioni possono ricevere messaggi di log o coda IBM MQ e agire su di essi in modo appropriato per le proprie esigenze aziendali.

Per ulteriori informazioni sulle API (application programming interface) disponibili, consultare [Sviluppo di applicazioni](#).

Configurazione dei gestori code dell'agent MFT

In seguito all'installazione, eseguire lo script `agent_name_create.mqsc` nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` per eseguire la configurazione necessaria per il gestore code dell'agent. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code dell'agent.

Procedura

1. Creare le code di operazione dell'agent.

Queste code sono denominate:

- SYSTEM.FTE.COMMAND.*nome_agent*
- SYSTEM.FTE.DATA.*nome_agent*
- SYSTEM.FTE.EVENT.*nome_agent*
- SYSTEM.FTE.REPLY.*nome_agent*
- SYSTEM.FTE.STATE.*nome_agent*

Per informazioni sui parametri della coda e su come vengono utilizzate le code, consultare [MFT Impostazioni della coda agent](#).

2. Creare le code di autorizzazioni agent.

Queste code sono denominate:

- SYSTEM.FTE.AUTHADM1.*nome_agent*
- SYSTEM.FTE.AUTHAGT1.*nome_agent*
- SYSTEM.FTE.AUTHMON1.*nome_agent*
- SYSTEM.FTE.AUTHOPS1.*nome_agent*
- SYSTEM.FTE.AUTHSCH1.*nome_agent*
- SYSTEM.FTE.AUTHTRN1.*nome_agent*

Per informazioni sui parametri della coda e su come vengono utilizzate le code, consultare [MFT Impostazioni della coda agent](#).

Operazioni successive

Per informazioni sulla creazione e la configurazione di un agent bridge di protocollo, consultare [fteCreateBridgeAgent \(creare e configurare un agent bridge di protocollo MFT\)](#) e [Configurazione di un bridge di protocollo per un server FTPS](#).

Concetti correlati

[“Connessione a IBM MQ” a pagina 787](#)

Tutte le comunicazioni di rete con i gestori code IBM MQ, incluse le comunicazioni correlate a Managed File Transfer, riguardano i canali IBM MQ. Un canale IBM MQ rappresenta un'estremità di un collegamento di rete. I canali sono classificati come canali di messaggi o canali MQI.

[“Configurazione di un gestore code a più istanze per utilizzare MFT” a pagina 793](#)

IBM WebSphere MQ 7.0.1 in poi supporta la creazione di gestori code a più istanze. Un gestore code a più istanze si riavvia automaticamente su un server standby. Managed File Transfer supporta la connessione ai gestori code dell'agent a più istanze, a un gestore code di coordinamento a più istanza e a un gestore code di comando a più istanze.

Attività correlate

“Configurazione dei gestori code di rete MFT” a pagina 788

Se la propria rete Managed File Transfer include più di un gestore code IBM MQ , tali gestori code IBM MQ devono essere in grado di comunicare in remoto tra loro.

“Configurazione del gestore code di coordinamento per MFT” a pagina 790

Dopo aver eseguito il comando **fteSetupCoordination** , eseguire lo script `coordination_qmgr_name.mqsc` nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` per eseguire la configurazione necessaria per il gestore code di coordinamento. Tuttavia, se si desidera eseguire questa configurazione manualmente, completare la seguente procedura sul gestore code di coordinamento.

Riferimenti correlati

[MFT Impostazioni coda agent](#)

[Coordinamento fteSetup](#)

Configurazione di un agent MFT per più canali in un cluster

Se si desidera utilizzare il supporto multicanale IBM MQ in una configurazione con cluster, impostare prima la proprietà **agentMultipleChannelsEnabled** su `true` e quindi completare i passi riportati in questo argomento.

Informazioni su questa attività

In un cluster, il supporto multicanale è abilitato dalle definizioni IBM MQ solo sul gestore code dell'agent di destinazione.

È necessario completare i passi in questo argomento in aggiunta ai passi di configurazione standard di IBM MQ richiesti per un agente Managed File Transfer , elencati in [“Configurazione di MFT per il primo utilizzo”](#) a pagina 786.

I seguenti esempi di configurazione utilizzano comandi **runmqsc** .

Procedura

1. Definire un canale ricevente del cluster per ogni canale che si desidera utilizzare. Ad esempio, se si utilizzano due canali:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
```

dove:

- *DESTQMGRNAME* è il nome del gestore code dell'agente di destinazione.
- *MFTCLUSTER* è il nome del cluster IBM MQ .

Si consiglia di utilizzare la convenzione di denominazione *MFTCLUSTER.DESTMGRNAME_n* per i canali, ma questa convenzione non è obbligatoria.

2. Definire un alias del gestore code corrispondente a ciascun canale. Ad esempio:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
```

È necessario utilizzare *SYSTEM.FTE.DESTQMGRNAME_n* convenzione di denominazione per [alias del gestore code](#) poiché l'agent di invio ricerca gli alias del gestore code di questo formato. I numeri

utilizzati per n devono iniziare da 1 ed essere consecutivi. È necessario rendere le definizioni a livello di cluster in modo che siano disponibili sul gestore code dell'agente di origine.

Affinché l'agent di origine e l'agent di destinazione determinino correttamente il numero di alias del gestore code, **non** definire un XMITQ predefinito per il gestore code.

Attività correlate

“Configurazione di un agent MFT per più canali: non in cluster” a pagina 799

Se si desidera utilizzare il supporto multicanale IBM MQ in una configurazione non in cluster, impostare la proprietà `agentMultipleChannelsEnabled` su `true` e completare i passi in questo argomento.

Riferimenti correlati

Il file MFT `agent.properties`

Configurazione di un agent MFT per più canali: non in cluster

Se si desidera utilizzare il supporto multicanale IBM MQ in una configurazione non in cluster, impostare la proprietà `agentMultipleChannelsEnabled` su `true` e completare i passi in questo argomento.

Informazioni su questa attività

In una configurazione non cluster, il supporto multicanale è abilitato dalle definizioni IBM MQ sul gestore code dell'agent di origine e di destinazione.

È necessario completare i passi in questo argomento in aggiunta ai passi di configurazione standard di IBM MQ richiesti per un agente Managed File Transfer, elencati in “Configurazione di MFT per il primo utilizzo” a pagina 786.

La seguente procedura presuppone che i canali mittente - destinatario vengano utilizzati per comunicare tra i gestori code di origine e di destinazione.

I seguenti esempi di configurazione utilizzano comandi **runmqsc**.

Procedura

1. Sul gestore code dell'agent di destinazione, definire un canale ricevente per ciascun canale che si desidera utilizzare. Ad esempio, se si utilizzano due canali:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

dove: `DESTQMGRNAME` è il nome del gestore code dell'agent di destinazione.

Si consiglia di utilizzare `TO.DESTMGRNAME_n` convenzione di denominazione per i canali, ma questa convenzione non è obbligatoria. I nomi dei canali riceventi devono corrispondere ai canali mittenti corrispondenti sul gestore code dell'agente di origine.

2. Sul gestore code dell'agente di origine, definire una coda di trasmissione per ciascun canale che si desidera utilizzare. Ad esempio, se si utilizzano due canali:

```
DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
```

Si consiglia di utilizzare la convenzione di denominazione `DESTMGRNAME_n` per code di trasmissione, ma questa convenzione non è obbligatoria. Le code di trasmissione definite vengono indicate dalle definizioni di canale mittente e dalle definizioni di alias del gestore code nei seguenti passi.

3. Sul gestore code dell'agent di origine, definire un canale mittente per ciascun canale che si desidera utilizzare. Ad esempio, se si utilizzano due canali:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_1)
```

```
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_2)
```

Si consiglia di utilizzare TO.DESTMGRNAME_n convenzione di denominazione per i canali, ma questa convenzione non è obbligatoria. I nomi dei canali mittente devono corrispondere ai canali riceventi corrispondenti sul gestore code dell'agente di destinazione.

4. Sul gestore code dell'agente di origine, definire un alias del gestore code corrispondente a ciascun canale. Ad esempio:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

È necessario utilizzare SYSTEM.FTE.DESTQMGRNAME_n convenzione di denominazione per gli alias del gestore code poiché l'agente mittente ricerca gli alias del gestore code di questo formato. I numeri che si utilizzano per *n* devono iniziare da 1 ed essere consecutivi.

Perché l'agent determini correttamente il numero di alias del gestore code, **non** definire un XMITQ predefinito per il gestore code.

Attività correlate

[“Configurazione di un agent MFT per più canali in un cluster” a pagina 798](#)

Se si desidera utilizzare il supporto multicanale IBM MQ in una configurazione con cluster, impostare prima la proprietà **agentMultipleChannelsEnabled** su `true` e quindi completare i passi riportati in questo argomento.

Riferimenti correlati

[Il file MFT agent.properties](#)

Configurazione degli agenti MFT con MSCS

L'impostazione Managed File Transfer (MFT) agent Microsoft Cluster Service (MSCS) è supportata, se la piattaforma è una supportata da MFT ed esegue una delle versioni di Windows.

Informazioni su questa attività

Questa attività descrive due scenari che possono essere seguiti per ottenere il failover di un agent MFT :

- Scenario 1: configurazione dell'agent come risorsa MSCS.
- Scenario 2: configurazione del gestore code dell'agent e dell'agent come risorse MSCS.

Procedura

Scenario 1: configurazione dell'agent come risorsa MSCS

- Per configurare l'agent come risorsa MSCS, completare la seguente procedura:
 - a) Installare Managed File Transfer localmente su ogni macchina nel cluster.
Consultare [Installazione di Managed File Transfer](#).
 - b) Creare l'agente sulla macchina primaria nel cluster.
L'agente deve essere configurato per connettersi al gestore code dell'agente utilizzando il trasporto CLIENT. Assicurarsi di creare tutti gli oggetti sul gestore code per questo agente. Per informazioni su come eseguire questa operazione, consultare [Impostazione dell'agente](#).
 - c) Modificare l'agent in modo che venga eseguito come servizio Windows e configurarlo in modo che non venga avviato automaticamente quando Windows viene riavviato impostando il campo **Tipo di avvio** per il servizio agent nello strumento Windows Servizi su Manuale.
Per ulteriori informazioni, vedi [Avvio di un agent MFT come servizio Windows](#).
 - d) Ripetere il passo [“2” a pagina 800](#) e il passo [“3” a pagina 800](#) dello scenario 1 sulla macchina secondaria.

Ciò garantisce che la struttura del file per i log, le proprietà e così via, esista sull'altra macchina nel cluster. Notare che non è necessario creare gli oggetti del gestore code come nel passo [“2” a pagina 800](#).

- e) Sulla macchina primaria, aggiungere l'agent come 'Servizio generico ' sotto il controllo di MSCS.
Per far ciò:

- a. Fare clic con il pulsante destro del mouse e selezionare **Ruolo -> Aggiungi risorsa -> 'Servizio generico'**.
- b. Dall'elenco dei servizi Windows , selezionare il servizio agent e completare la procedura guidata di configurazione facendo clic su **Avanti**.

Il servizio agent è ora aggiunto come risorsa MSCS. Se si verifica un failover, il servizio agent verrà avviato sull'altra macchina.

Scenario 2: configurazione del gestore code dell'agent e dell'agent come risorse MSCS

- Per configurare il gestore code dell'agent e l'agent come risorse MSCS, completare la seguente procedura:

- a) Configurare il gestore code dell'agente da eseguire come risorsa MSCS.

Per informazioni su come svolgere questa procedura, consultare [“Inserimento di un gestore code sotto il controllo di MSCS” a pagina 512](#).

- b) Creare l'agente sulla macchina primaria nel cluster.

L'agent deve essere configurato per connettersi al gestore code dell'agent utilizzando il trasporto BINDINGS. Assicurarsi di creare tutti gli oggetti sul gestore code per questo agente. Per informazioni su come eseguire questa operazione, consultare [Impostazione dell'agente](#).

- c) Modificare l'agent in modo che venga eseguito come servizio Windows e configurarlo in modo che non venga avviato automaticamente quando Windows viene riavviato impostando il campo **Tipo di avvio** per il servizio agent nello strumento Windows Servizi su Manuale.

Per ulteriori informazioni, vedi [Avvio di un agent MFT come servizio Windows](#).

- d) Controllare che il gestore code dell'agent (sotto il controllo MSCS) sia in esecuzione sulla macchina secondaria.

L'agent creato su questa macchina si conatterà al gestore code utilizzando il trasporto BINDINGS, pertanto deve essere disponibile quando l'agent viene creato.

- e) Ripetere il passo [“2” a pagina 801](#) e il passo [“3” a pagina 801](#) dello scenario 2 sulla macchina secondaria.

Ciò garantisce che la struttura del file per i log, le proprietà e così via, esista sull'altra macchina nel cluster. Notare che non è necessario creare gli oggetti del gestore code come nel passo [“2” a pagina 801](#).

- f) Aggiungere l'agent come 'Servizio generico ' nel controllo MSCS.

Per far ciò:

- a. Fare clic con il pulsante destro del mouse e selezionare **Ruolo -> Aggiungi risorsa -> 'Servizio generico'**.
- b. Dall'elenco dei servizi Windows , selezionare il servizio agent e completare la procedura guidata di configurazione facendo clic su **Avanti**.

- g) Modificare le proprietà della risorsa del servizio agent per aggiungere la risorsa del gestore code nell'elenco delle dipendenze.

Ciò garantisce che la risorsa del gestore code venga avviata prima dell'avvio dell'agente.

- h) Portare la risorsa del gestore code non in linea, quindi portare la risorsa agent in linea. Verificare se sia la risorsa del gestore code che l'agent sono avviati.

Se si verifica un failover, il servizio agent e il gestore code agent verranno avviati sulla macchina secondaria.

Agent ad alta disponibilità in Managed File Transfer

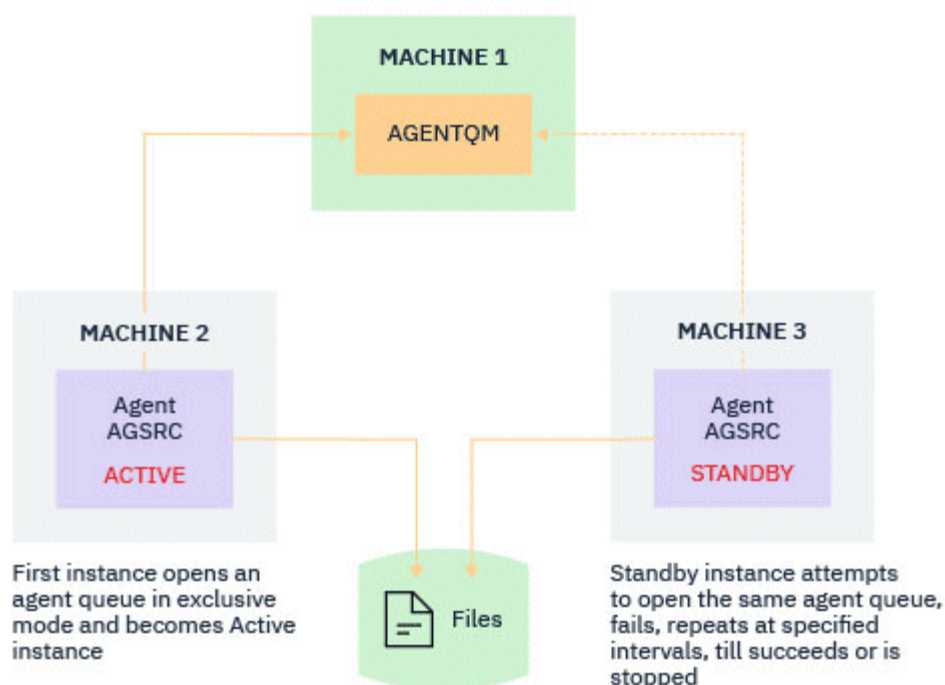
È possibile configurare gli agent bridge o standard in MFT per l'esecuzione in una configurazione HA (high availability). Una coppia di istanze agent con configurazioni identiche sono coinvolte nella configurazione HA, dove un'istanza è in esecuzione su una macchina mentre un'altra è in esecuzione su una macchina diversa. Entrambe le istanze sono configurate per connettersi allo stesso gestore code dell'agent.

Panoramica

Solo una delle due istanze, denominata *istanza attiva*, sta elaborando i trasferimenti file, mentre l'altra istanza, denominata *istanza standby*, si trova in uno stato parzialmente inizializzato e non può elaborare alcun trasferimento file.

Quando un'istanza attiva ha esito negativo o perde la connettività al gestore code, l'istanza in standby completa la sua inizializzazione, diventa attiva e avvia l'elaborazione dei trasferimenti file. Tutti i trasferimenti in corso quando l'istanza attiva non è riuscita vengono ripristinati dall'ultimo punto di controllo noto.

La seguente illustrazione mostra una configurazione comune degli agent attivi e



standby:

Note:

1. Un'istanza di un agente è in esecuzione su due macchine differenti, con una delle istanze come *istanza attiva* e l'altra come *istanza standby*.
2. Ogni istanza dell'agente è in esecuzione su una macchina diversa, con una delle istanze come istanza attiva e l'altra come istanza standby.
3. La stessa serie di code agent è condivisa tra entrambe le istanze dell'agent.
4. Entrambe le istanze dell'agent devono accedere allo stesso filesystem condiviso per eseguire i trasferimenti gestiti.

Il meccanismo dell'istanza dell'agent standby attivo funziona utilizzando un blocco su una risorsa condivisa. L'istanza dell'agente che prende un blocco sulla risorsa condivisa diventa l'istanza attiva mentre l'altra istanza (che non riesce a prendere un blocco) diventa un'istanza standby.

La risorsa condivisa è una nuova coda, SYSTEM.FTE.HA.<agent name>. Questa coda viene creata automaticamente quando viene configurato un agent IBM MQ 9.1.4 o successivo.

Come funziona il processo

Per creare un agent HA, si crea un agent con parametri di configurazione identici su due macchine eseguendo il comando **fteCreateAgent** o **fteCreateBridgeAgent** utilizzando il parametro **-x** aggiuntivo insieme alla proprietà dell'agent **highlyAvailable** nel file `agent.properties` impostato su `true`.

Note:

- Entrambe le configurazioni devono puntare allo stesso gestore code dell'agente.
- Le code agent richieste devono essere create solo una volta sul gestore code agent.

Per ulteriori informazioni sulla proprietà dell'agent **highlyAvailable**, consultare il comando **fteCreateAgent** per ulteriori informazioni sul parametro **-x** e il file `agent.properties`.

Nota: L'esecuzione del comando **fteCreateAgent** o **fteCreateBridgeAgent** crea un file MQSC contenente gli script richiesti per creare gli oggetti IBM MQ nel gestore code dell'agent e nella coda `SYSTEM.FTE.HA.agent name`. Questo file MQSC viene creato indipendentemente dal fatto che si specifichi o meno il parametro **-x**.

Durante la creazione di una configurazione dell'agent ad alta disponibilità, il comando **fteCreateAgent** o **fteCreateBridgeAgent** verifica l'esistenza di un'istanza dello stesso agent presente altrove sottoscrivendo l'argomento `SYSTEM.FTE/Agents/agent name`. Se viene trovata un'istanza dello stesso agent, il comando crea la configurazione richiesta sul file system ma non pubblica nuovamente la creazione dell'agent.

Quando un agent viene avviato in modalità HA:

1. L'agent tenta di aprire la coda `SYSTEM.FTE.HA.agent name` in modalità GET esclusiva.
2. Se l'agent apre correttamente la coda `SYSTEM.FTE.HA.agent name`, questa diventa l'*istanza attiva* di un agent e un ulteriore processo di avvio continua.
3. Se il tentativo di aprire la coda `SYSTEM.FTE.HA.agent name` in una modalità GET esclusiva ha esito negativo con il codice motivo `MQRC_OBJECT_IN_USE`, significa che esiste già un'istanza attiva dell'agent in esecuzione altrove. Di conseguenza, questa istanza diventa l'*istanza standby* dell'agente.

L'istanza in standby tenta di aprire la coda `SYSTEM.FTE.HA.agent name` a intervalli specificati. Una proprietà agent aggiuntiva **standbyPollInterval** viene fornita per questo scopo nel file `agent.properties`.

Con il valore predefinito, l'istanza in standby tenta di aprire la coda `SYSTEM.FTE.HA.agent name` ogni cinque secondi. Ciò si ripete fino a quando l'istanza riesce ad aprire la coda `SYSTEM.FTE.HA.agent name` o viene arrestata utilizzando il comando **fteStopAgent**.

La proprietà **standbyPollInterval** viene utilizzata anche da tutte le istanze per stabilire il tempo di attesa di un'istanza tra i tentativi di riconnessione se viene disconnessa dal gestore code dell'agent.

Più istanze in standby

Tutte le istanze standby tentano di prendere la coda `SYSTEM.FTE.HA.agent name` in una modalità GET esclusiva e l'istanza che ha esito positivo, dopo che l'istanza attiva ha avuto esito negativo, diventa l'istanza attiva.

L'istanza attiva conserva le informazioni di tutte le istanze di standby note e pubblica le informazioni come parte della pubblicazione dello stato agent. L'output del comando **fteShowAgentDetails**, la risposta GET REST API dell'agent e il plug-in IBM MQ Explorer MFT visualizzano le informazioni su tutte le istanze in standby.

Per ulteriori informazioni, vedi gli output di esempio del comando **fteShowAgentDetails** e la risposta GET REST API dell'agent.

Consultare [MFT Messaggi di stato dell'agente](#) per esempi di informazioni sullo stato dell'agente in formato XML.

Requisito versione

Gli agent attivi e standby devono essere IBM MQ 9.1.4 o superiori.



Attenzione:

- Non è possibile configurare o avviare le versioni di IBM MQ precedenti a IBM MQ 9.1.4 in modalità alta disponibilità.
- Sia le istanze attive che quelle in standby devono eseguire la stessa versione del codice.

La versione delle istanze attive e standby viene convalidata per garantire che entrambe le istanze siano della stessa versione. Una coda dinamica temporanea viene utilizzata per la comunicazione tra le istanze. Due proprietà agent, **dynamicQueuePrefix** e **modelQueueName**, definite nel file `agent.properties`, generano il nome della coda dinamica temporanea.

Informazioni richieste per gli agent ad alta disponibilità in Managed File Transfer

Esistono vari tipi di informazioni che è necessario conoscere sugli agent standard o bridge MFT in esecuzione in una configurazione ad alta disponibilità. Queste informazioni includono i diversi metodi con cui viene avviato l'agent, come identificare l'istanza dell'agent nel file di log e le informazioni sullo stato per l'agent.

Avvio di un agent

Un'istanza di un agent è in esecuzione in una modalità non HA altrove

Se viene effettuato un tentativo di avviare un'altra istanza dell'agent non configurata come un agent HA, viene prima effettuato un controllo per verificare se è possibile acquisire un blocco sulla coda `SYSTEM.FTE.HA.agent name`.

Poiché l'altra istanza è stata avviata in modalità non HA, il vincolo sulla coda `SYSTEM.FTE.HA.agent name` verrà acquisito da questa istanza. L'agent continua l'inizializzazione, ma non riesce in un secondo momento perché la coda comandi viene aperta esclusivamente da un'altra istanza.

In questo caso, i messaggi mostrati nel seguente esempio vengono registrati sul file `output0.log` dell'agente e l'agente continua il suo tentativo di aprire la coda comandi ogni 30 secondi:

```
BFGMQ1045I: Coda di sistema dell'agent 'SYSTEM.FTE.COMMAND.SRC' è configurato come NOSHARE o DEFSOPT (CONDIVISO).
```

```
BFGAG0035W: L'agente ha ricevuto il codice motivo MQI 2042 durante il tentativo di apertura della coda 'SYSTEM.FTE.COMMAND.SRC' sul gestore code 'MFTHAQM' con nome connessione 'localhost (1414)' e canale 'MFT_HA_CHN'. L'agent tenterà nuovamente l'operazione ogni 30 secondi.
```

Un'istanza di un agent è in esecuzione in una modalità HA altrove

Se viene effettuato un tentativo di avviare un'altra istanza dell'agent non configurata come un agent HA, viene prima effettuato un controllo per verificare se è possibile acquisire un blocco sulla coda `SYSTEM.FTE.HA.agent name`.

Poiché l'altra istanza è stata eseguita come istanza attiva, il tentativo di acquisire un blocco ha esito negativo. L'istanza non viene avviata e il seguente errore viene registrato nel file `output0.log` dell'agent:

```
BFGAG0194E: Un'istanza di questo agent è già in esecuzione altrove. Pertanto, questa istanza non può continuare e verrà terminata.
```



Avvio dell'agent come servizio Windows

Su Windows, è possibile avviare un agent come servizio Windows.

Durante l'avvio, Windows avvia l'agent MFT in modalità normale o HA. Se l'agent è configurato per essere eseguito in modalità HA, il servizio viene eseguito come istanza attiva o standby, a seconda di quale istanza acquisisce per prima il blocco.

Identificazione del tipo di istanza di un agente nel file di log

I messaggi informativi vengono scritti nel file `output0.log` dell'agente per indicare il tipo di istanza. Quando un'istanza agent viene avviata come istanza attiva, viene scritto il seguente messaggio:

```
BFGAG0193I: L'agent è stato inizializzato correttamente come istanza attiva.
```

Quando un'istanza dell'agent viene avviata come istanza standby, viene scritto il seguente messaggio:

```
BFGAG0193I: L'agent è stato inizializzato correttamente come un'istanza standby.
```

Aggiornamenti dello stato dell'agente

Poiché esistono due istanze dello stesso agent in esecuzione, è necessario disporre delle informazioni su entrambe le istanze nella pubblicazione dello stato dell'agent.

Si noti che l'istanza attiva è quella che pubblica lo stato di entrambe le istanze.

Istanza standby

Durante la pubblicazione dello stato dell'agent, l'istanza attiva controlla la durata della pubblicazione dell'istanza in standby.

Esistono due proprietà aggiuntive nel file `agent.properties` per questo scopo:

- **standbyStatusExpiry** è la scadenza per il messaggio di stato standby da inserire nella coda comandi dell'agent. Il messaggio scade se l'istanza attiva di un agente non elabora questo messaggio in tale periodo.

Per impostazione predefinita, il valore di **standbyStatusExpiry** è 30 secondi. Il messaggio è anche un messaggio con priorità bassa, 9, per consentire l'elaborazione con priorità delle richieste di trasferimento rispetto ai messaggi di stato standby.

- **standbyStatusPublishInterval** imposta la frequenza con cui l'istanza in standby pubblica il relativo stato.

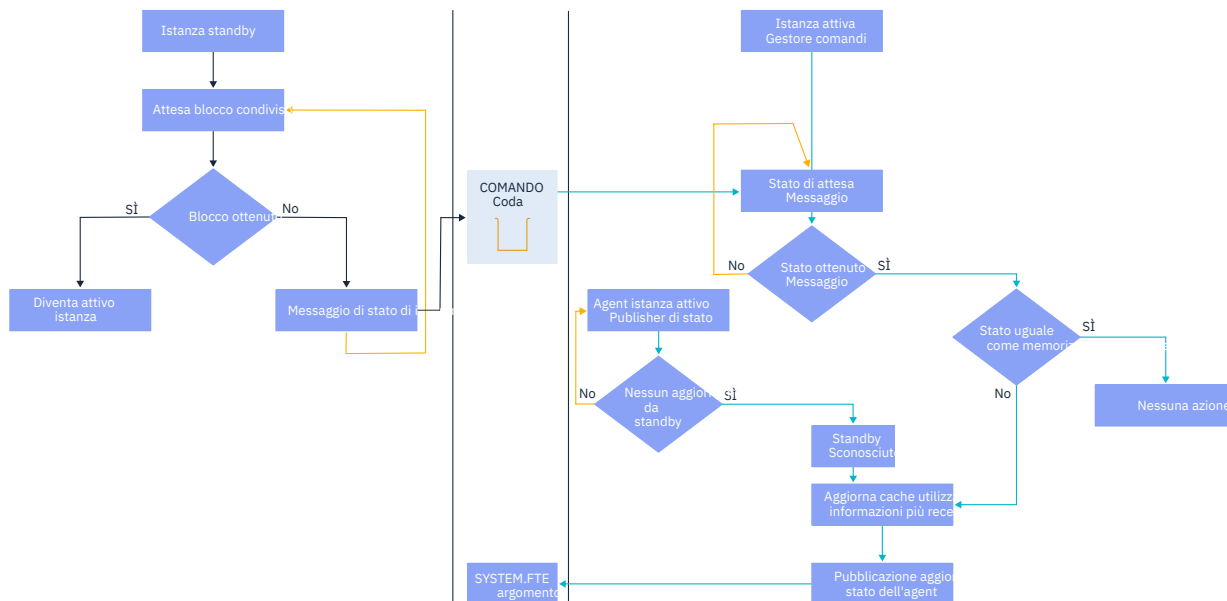
Istanza attiva

L'istanza attiva effettua le seguenti operazioni per elaborare gli aggiornamenti dello stato dall'istanza standby:

1. Richiama il messaggio dalla coda `SYSTEM.FTE.COMMAND.<agent name>` e delega l'elaborazione del messaggio a un thread di lavoro.
2. Il thread di lavoro richiama il contenuto dal corpo del messaggio, aggiorna l'oggetto di stato dell'agent con le informazioni sull'istanza in standby e notifica al publisher di stato agent di pubblicare lo stato.
3. Il publisher dello stato dell'agent pubblica lo stato.

Notare che le ottimizzazioni vengono eseguite qui per memorizzare nella cache le informazioni sullo stato di standby. Quando viene effettuata una richiesta, il publisher dello stato dell'agent controlla il nuovo stato con lo stato memorizzato nella cache e pubblica solo se è presente una differenza.

Il seguente diagramma descrive il flusso seguito dalle istanze attive o standby per pubblicare lo stato di un agent:



Eliminazione di istanze, failover e manutenzione in agent altamente disponibili

Le istanze Managed File Transfer ad alta disponibilità possono essere eliminate, possono avere esito negativo in vari modi e potrebbero richiedere manutenzione.

Eliminazione dello stato dell'istanza in standby

Ci possono essere situazioni in cui l'istanza attiva è occupata con i trasferimenti e non è in grado di elaborare i messaggi di stato dell'istanza in standby, l'istanza in standby non è riuscita o non sta pubblicando i messaggi di stato per qualsiasi motivo.

In tali scenari, l'agente attivo che era consapevole della presenza di un'istanza standby attende il valore specificato dalla proprietà **standbyStatusDiscardTime** del file `agent.properties` prima di rimuovere l'istanza standby dal relativo elenco. Il valore predefinito per questa proprietà è 600 secondi, che è il doppio della proprietà **standbyStatusPublishInterval**.

Failover normale su un'istanza

È necessario utilizzare il comando **fteStopAgent** con l'opzione **-i** per eseguire un normale failover.

Ciò garantisce che l'istanza attiva venga arrestata immediatamente. Se si arresta un agent senza l'opzione **-i**, l'agent continua l'esecuzione fino a quando tutti i trasferimenti in corso non vengono completati dall'istanza attiva, quindi, il failover potrebbe richiedere molto tempo.

Eventuali trasferimenti in volo riprendono dall'ultimo punto di controllo noto.

Failover di un'istanza in altre situazioni

Se un'istanza attiva termina in un modo non normale o se l'intera macchina ha esito negativo, la connessione alla coda dell'agent viene interrotta e il gestore code chiude tutte le code aperte, inclusa la coda `SYSTEM.FTE.HA.<agent name>` e le connessioni.

A causa di ciò, l'istanza standby acquisisce il GET esclusivo e completa il resto dell'inizializzazione dell'agent.

Anche in questo caso, i trasferimenti in volo riprendono dagli ultimi punti di controllo noti.

In caso di interruzione di una connessione al gestore code

Modalità client

Un processo agent è composto da diversi thread. Oltre ai thread predefiniti, ad esempio, un thread che pubblica lo stato dell'agente a intervalli regolari, ogni richiesta di trasferimento viene gestita con una serie di thread che terminano dopo il completamento di un trasferimento.

Molti di questi thread si collegano al gestore code dell'agent e inviano e ricevono messaggi. È possibile che una di queste connessioni possa interrompersi a causa di un problema di rete o di un errore del gestore code. Quando un thread rileva un problema di connessione interrotta, il thread informa il thread principale di avviare il ripristino e termina.

Il thread principale avvia quindi un altro thread per attendere che venga stabilita una connessione al gestore code. Una volta riconnesso, viene effettuato un tentativo di acquisire il GET esclusivo per l'agente. Se l'operazione ha esito positivo, l'agent continua a completare il recupero e diventa l'istanza attiva. Se il tentativo di acquisire il GET esclusivo non riesce, l'istanza diventa standby.

Modalità bind

Durante la connessione in modalità bind, se un agent perde la connessione, il processo dell'agent termina. Il controller processi gestisce il riavvio dell'agent. Quando un agent viene riavviato, viene eseguito il tentativo di acquisire il GET esclusivo per se stesso.

Se l'agent ha esito positivo, diventa un'istanza attiva; in caso contrario, l'agent diventa un'istanza di standby.

Applicazione degli aggiornamenti del livello di manutenzione

La procedura per l'applicazione della manutenzione agli agent ad alta disponibilità è simile a quella documentata per i gestori code a più istanze. Per ulteriori informazioni, consultare [Applicazione degli aggiornamenti del livello di manutenzione ai gestori code a più istanze su Windows](#) o [Applicazione degli aggiornamenti del livello di manutenzione ai gestori code a più istanze su AIX](#) oppure [Applicazione degli aggiornamenti del livello di manutenzione ai gestori code a più istanze su Linux](#).

È necessario arrestare l'agent in esecuzione sulla macchina su cui deve essere applicato il livello di manutenzione, prima di applicare la manutenzione. Se si sta aggiornando un'istanza attiva, per la continuità dei trasferimenti, è necessario eseguire il failover dell'istanza attiva su un'istanza in standby.

Una volta completato l'aggiornamento, è possibile avviare l'istanza dell'agent, eseguire il failover dell'istanza attiva corrente sull'istanza aggiornata e quindi aggiornare l'istanza standby.


Migrazione di agent da una versione precedente del prodotto

Gli agent migrati dalle versioni di IBM MQ precedenti a IBM MQ 9.1.4 vengono eseguiti come non altamente disponibili. È possibile eseguirli in modalità alta disponibilità seguendo la procedura in [Migrazione di agent Managed File Transfer da una versione precedente](#).

Configurazione di un programma di registrazione MFT

Quando Managed File Transfer trasferisce i file, pubblica le informazioni sulle sue azioni in un argomento sul gestore code di coordinamento. Il programma di registrazione database è un componente facoltativo di Managed File Transfer che è possibile utilizzare per copiare queste informazioni in un database per scopi di analisi e controllo.

Esistono tre versioni del logger:

-  logger di file autonomo
- programma di registrazione database autonomo
- logger Java Platform, Enterprise Edition (Java EE)

Logger su IBM i

IBM i

I logger Managed File Transfer non sono supportati sulla piattaforma IBM i .

Programma di registrazione file autonomo

ALW

Il programma di registrazione file autonomo è un processo Java che viene eseguito sul sistema su cui è presente il gestore code di coordinamento o su un sistema su cui è presente un gestore code con connettività al gestore code di coordinamento. Il programma di registrazione file autonomo utilizza i bind IBM MQ per connettersi al gestore code associato. Il programma di registrazione autonomo viene creato utilizzando il comando **fteCreateLogger** .

Windows È possibile eseguire il programma di registrazione file autonomo come servizio Windows per garantire che il programma di registrazione file continui l'esecuzione quando si scollega dalla sessione Windows e può essere configurato per essere avviato automaticamente al riavvio di un sistema. Per ulteriori informazioni, consultare [“Installazione del programma di registrazione file autonomo MFT” a pagina 809](#).

Il programma di registrazione file autonomo non è supportato sulle piattaforme riportate di seguito:

- **z/OS** z/OS
- **IBM i** IBM i

Programma di registrazione database autonomo

Il programma di registrazione database autonomo è un'applicazione Java che viene installata su un sistema che contiene un gestore code e un database. Il programma di registrazione database autonomo è spesso installato sullo stesso sistema del gestore code di coordinamento, tuttavia può essere installato anche sullo stesso sistema di qualsiasi gestore code che abbia la connettività al gestore code di coordinamento. Il programma di registrazione database autonomo utilizza i bind IBM MQ per collegarsi al gestore code associato e un driver JDBC di tipo 2 o 4 per connettersi a un database Db2 o Oracle . Questi tipi di connessione sono richiesti perché il programma di registrazione database autonomo utilizza il supporto XA del gestore code per coordinare una transazione globale sia sul gestore code che sul database, proteggendo i dati.

Windows Se si utilizza un sistema Windows , è possibile eseguire i logger autonomi come servizi Windows per garantire che i logger continuino l'esecuzione quando ci si scollega dalla sessione Windows . Per ulteriori informazioni, consultare [“Installazione del programma di registrazione database autonomo MFT” a pagina 817](#) per un programma di registrazione database autonomo.

Programma di registrazione database Java EE

Il programma di registrazione del database Java EE viene fornito come file EAR, che viene installato in un server delle applicazioni. Ciò può essere più conveniente rispetto all'utilizzo del programma di registrazione database autonomo se si dispone di un ambiente del server delle applicazioni Java EE esistente, poiché il programma di registrazione database Java EE può essere gestito insieme alle altre applicazioni enterprise. È anche possibile installare il programma di registrazione database Java EE su un sistema separato sui sistemi che ospitano il server IBM MQ e il database. Il programma di registrazione database Java EE viene supportato per l'utilizzo con i database Db2 e Oracle . Il programma di registrazione database Java EE supporta anche Oracle Real Application Clusters quando installato su WebSphere Application Server 7.0.

Per istruzioni su come configurare un programma di registrazione, consultare i seguenti argomenti:

- [“Installazione del programma di registrazione file autonomo MFT” a pagina 809](#)
- [“Installazione del programma di registrazione database autonomo MFT” a pagina 817](#)

- [“Installazione del programma di registrazione database Java EE per MFT” a pagina 822](#)

Attività correlate

[“Utilizzo di MFT con un database remoto” a pagina 818](#)

È possibile utilizzare il programma di registrazione Managed File Transfer per comunicare con un database su un sistema remoto.

Riferimenti correlati

[Gestione degli errori del programma di registrazione MFT e rifiuto dei messaggi](#)

[Proprietà di configurazione del programma di registrazione MFT](#)

Installazione del programma di registrazione file autonomo MFT

Il programma di registrazione file autonomo è un processo Java che può connettersi a un gestore code di coordinamento utilizzando la modalità di bind IBM MQ o la modalità client. Per definire un programma di registrazione file autonomo, utilizzare il comando **fteCreateLogger** e seguire i passi in questo argomento.

Informazioni su questa attività

Per ulteriori informazioni sul programma di registrazione file autonomo, consultare [“Configurazione di un programma di registrazione MFT” a pagina 807](#). I passaggi in questo argomento configurano un programma di registrazione per la connessione a un gestore code di coordinazione. Per le configurazioni del logger alternativo, consultare [“Configurazioni alternative per un logger autonomo MFT” a pagina 820](#).

Il programma di registrazione file autonomo non è supportato sulle piattaforme riportate di seguito:

-  z/OS
-  IBM i

Procedura

1. Assicurarsi di avere installato il componente Managed File Transfer Logger . Per ulteriori informazioni, consultare [Opzioni del prodotto Managed File Transfer](#).
2. Eseguire il comando **fteCreateLogger** specificando il gestore code di coordinamento e impostando il parametro **-loggerType** su FILE per creare il programma di registrazione file autonomo.
Per ulteriori informazioni, vedi [fteCreateLogger](#).
3. Opzionale: Se si desidera utilizzare un formato personalizzato, è possibile modificare il file XML creato dal comando **fteCreateLogger** . La definizione del formato di log si trova nel file `FileLoggerFormat.xml` .
Per ulteriori informazioni, consultare [“Formato del programma di registrazione file autonomo MFT” a pagina 810](#).
4. Eseguire i comandi MQSC, forniti dal comando **fteCreateLogger** , sul gestore code di coordinamento per creare le code del programma di registrazione.
5. Identificare un utente per eseguire il processo del logger e configurare le autorizzazioni per tale utente.
Per ulteriori informazioni, consultare [“Configurazione dell'accesso utente per un programma di registrazione file autonomo MFT” a pagina 816](#).
6. Opzionale: Configurare ulteriormente il programma di registrazione file autonomo modificando il file `logger.properties` creato quando è stato eseguito il comando **fteCreateLogger** .
Questo file è un file delle proprietà Java costituito da coppie chiave - valore. Il file `logger.properties` è nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` . Per ulteriori informazioni sulle proprietà disponibili e sui relativi effetti, vedere [MFT proprietà di configurazione del programma di registrazione](#).

7. **Windows**

Opzionale: Se si utilizza un sistema Windows , eseguire il programma di registrazione file autonomo come servizio Windows .

Eseguire il comando **fteModifyLogger** con il parametro **-s** . Per ulteriori informazioni, consultare [fteModifyLogger](#).

8. Avviare il programma di registrazione file autonomo con il comando **fteStartLogger** .

Per ulteriori informazioni, vedi [fteStartLogger](#).

Se hai eseguito il passo precedente e hai utilizzato il comando **fteModifyLogger** con il parametro **-s** su Windows, il programma di registrazione file autonomo viene avviato come servizio Windows .

9. Controllare l'output del programma di registrazione. Il programma di registrazione file autonomo genera due tipi di output, dati di controllo trasferimento file e dati di diagnostica del programma di registrazione.

I dati di controllo del trasferimento file sono disponibili in *MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs*. I dati diagnostici del programma di registrazione possono essere trovati in *MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name*.

10. Arrestare il programma di registrazione utilizzando il comando **fteStopLogger** .

Per ulteriori informazioni, vedi [fteStopLogger](#).

Risultati

Attività correlate

[“Configurazione dell'accesso utente per un programma di registrazione file autonomo MFT” a pagina 816](#)

In un ambiente di test, è possibile aggiungere qualsiasi nuovo privilegio necessario al proprio account utente normale. In un ambiente di produzione, si consiglia di creare un nuovo utente con le autorizzazioni minime richieste per eseguire il lavoro.

Riferimenti correlati

[Proprietà di configurazione del programma di registrazione MFT](#)

[Programma di registrazione fteStart\(avvia un programma di registrazione MFT \)](#)

[Programma di registrazione fteCreate\(crea un file MFT o un programma di registrazione database\)](#)

[fteModifyLogger \(esegue un logger MFT come un servizio Windows \)](#)

[fteStopLogger \(arresta un logger MFT \)](#)

[“Formato del programma di registrazione file autonomo MFT” a pagina 810](#)

Il formato delle informazioni del messaggio scritte dal programma di registrazione file può essere definito nel file `FileLoggerFormat.xml` .

[Autorità per il logger MFT](#)

ALW **Formato del programma di registrazione file autonomo MFT**

Il formato delle informazioni del messaggio scritte dal programma di registrazione file può essere definito nel file `FileLoggerFormat.xml` .

La directory di configurazione per il programma di registrazione si trova in *MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name*. Quando si crea un nuovo programma di registrazione file, viene creata una versione di questo file che contiene una serie predefinita di definizioni utilizzate dal programma di registrazione file. Per ulteriori informazioni sulla definizione del formato di log predefinito, consultare [MFT formato di log predefinito del programma di registrazione file autonomo](#).

Se si desidera specificare il proprio formato di log personalizzato, modificare il file `FileLoggerFormat.xml` .

Una definizione del formato di log personalizzato

Una definizione del formato di log è costituita da una serie di tipi di messaggi con ciascun tipo di messaggio che dispone di una definizione del formato. Una definizione di formato per un tipo di messaggio consiste in una serie di inserimenti forniti in formato XPATH e in un separatore utilizzato per separare ciascun inserimento. L'ordine degli inserimenti determina l'ordine in cui il contenuto viene inserito nelle linee generate per l'output nei file di log. Ad esempio, questa è la definizione per il tipo di messaggio `callStarted` :

```
<callStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/
        @agent</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@type</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@name</insert>
      <insert type="system" width="0" ignoreNull="true">callArguments</insert>
    </inserts>
    <separator></separator>
  </format>
</callStarted>
```

Questo formato produce una riga nel file di log simile alla seguente:

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02; [CSTR];
AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

Gli inserimenti forniti nella definizione del formato sono nell'ordine in cui le informazioni vengono visualizzate sulla riga del file di log. Per ulteriori informazioni sullo schema XML che definisce il formato per il file `FileLoggerFormat.xml` , consultare [Stand-alone file logger format XSD](#).

Tipi di messaggio

Gli agent FTE scrivono una serie di tipi di messaggi differenti nell'argomento secondario `SYSTEM.FTE/Log` . Per ulteriori informazioni, consultare [SYSTEM.FTE ArgomentoFTE](#). La definizione del file di log può contenere definizioni di formato per questi tipi di messaggi:

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

Il formato dei messaggi può variare. La maggior parte dei tipi di messaggi scrive una riga singola nel file di log per ogni messaggio di log utilizzato dall'argomento secondario `SYSTEM.FTE/Log` . Ciò porta al caso semplice in cui gli indirizzi XPATH forniti nella definizione del formato di log si riferiscono alla root del messaggio. Questi sono i tipi di messaggi che utilizzano questo metodo per scrivere l'output:

```

callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete

```

L'altro metodo utilizzato per scrivere un messaggio di log utilizza più righe per rappresentare gli elementi in una serie di trasferimenti all'interno di un messaggio di log. In questo caso, il formato fornito viene applicato a ciascun elemento nella serie di trasferimenti all'interno del messaggio di log. Se si desidera includere informazioni specifiche per ciascun elemento all'interno della serie di trasferimenti, è necessario che l'XPATH fornito utilizzi l'elemento come root XPATH. Questi sono i tipi di messaggi che utilizzano questo metodo per scrivere l'output:

```

scheduleSubmitTransferSet
transferProgress

```

Viene scritta una riga di output per ogni elemento nella serie di trasferimento. Le informazioni che si desidera correggere per tutti gli elementi in una serie di trasferimento possono ancora utilizzare gli indirizzi XPATH relativi al root del messaggio di log. Nel seguente esempio di definizione del formato `transferProgress` semplificato, sono corretti la data / ora e l'ID trasferimento. Tutte le informazioni relative a un elemento come root varieranno per ogni riga scritta. In questo esempio vengono scritte le informazioni sul file di origine e di destinazione per ogni elemento.

```

<transferProgress>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">source/file |
        source/queue</insert>
      <insert type="user" width="0" ignoreNull="false">source/file/@size |
        source/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file |
        destination/queue</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file/@size |
        destination/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
      <insert type="user" width="0" ignoreNull="true">status/supplement</insert>
    </inserts>
    <separator></separator>
  </format>
</transferProgress>

```

Ciò produce una voce del file di log di una o più righe in questo formato:

```

2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPRO];0
;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;

```

Inserisci formato

Sono disponibili due tipi di inserimento quando si definisce un formato per un tipo di messaggio: utente e sistema. Il tipo di un inserimento è definito nell'attributo `type` dell'elemento di inserimento. Entrambi i tipi di inserimenti possono anche avere il layout personalizzato utilizzando gli attributi **width** e **ignoreNull** dell'elemento di inserimento. Ad esempio:

```
<insert type="user" width="48" ignoreNull="false"/>/transaction/@ID</insert>
```

In questo esempio, l'inserimento prende le informazioni trovate nel messaggio di log in `/transaction/@ID` e le elimina o le riempisce di 48 caratteri prima di scriverle nel log. Se il contenuto di `/transaction/@ID` è null, scrive la stringa null dopo averla riempita con 48 caratteri perché l'attributo `ignoreNull` è impostato su `false`. Se `ignoreNull` è impostato su `true`, viene invece scritta la stringa vuota, riempita con 48 caratteri. L'impostazione `width="0"` indica che la larghezza della colonna non è ritagliata, non significa che la larghezza è ritagliata a 0. L'attributo `ignoreNull` può essere utilizzato in questo modo per rilevare nel log quando viene rilevato un valore null quando non era previsto. Ciò può essere utile quando si esegue il debug di una nuova definizione del file di log.

Inserimenti definiti dall'utente

Un inserimento utente contiene un indirizzo XPATH per le informazioni da scrivere in tale inserimento. Questo indirizzo fa riferimento a una parte di informazioni trovata nel messaggio di log FTE. Per ulteriori informazioni sui formati dei messaggi di log, consultare:

- [Formati dei messaggi del log di trasferimenti file](#)
- [Formati del messaggio di registrazione del trasferimento file pianificato](#)
- [Formato del messaggio di log del monitoraggio MFT](#)

Inserimenti definiti dal sistema

Gli inserimenti definiti dal sistema contengono una parola chiave che fa riferimento a una parte di informazioni che non è possibile trovare nel messaggio di log o che non è facile definire utilizzando il linguaggio XPATH.

Gli inserimenti di sistemi supportati sono:

- `type` - Scrive il tipo di messaggio di log in un formato breve.
- `callArguments` - Scrive la serie di argomenti forniti a una chiamata gestita in formato separato da spazi.
- `transferMetaData` - Scrive la serie di voci di metadati definite per un trasferimento in formato `chiave=value` separato da virgole.

La seguente tabella elenca il valore di "tipo" per gli inserimenti definiti dal sistema per ciascun tipo di messaggio.

Tipo messaggio	Valore dell'inserimento di sistema "tipo"
callCompleted	[CCOM]
callStarted	[CSTR]
monitorAction	[MACT]
monitorCreate	[MCRT]
monitorFired	[MFIR]

Tabella 51. Riepilogo dei tipi di messaggio supportati e relativi inserimenti di sistema "tipo". (Continua)

Tipo messaggio	Valore dell'inserimento di sistema "type"
notAuthorized	[AUTH]
scheduleDelete	[MODELLO]
scheduleExpire	[SEXP]
scheduleSkipped	[SSKP]
Informazioni scheduleSubmit	[SSIN]
Trasferimento scheduleSubmit	[SSTR]
scheduleSubmitTransferSet	[SSTS]
transferStarted	[TSTR]
transferCancelled	[TCAN]
transferComplete	[TCOM]
transferDelete	[DEL]
transferProgress	[TPRO]

Riferimenti correlati

[Formato di log predefinito del programma di registrazione file autonomo MFT](#)

[Formato del programma di registrazione file autonomo XSD](#)

SYSTEM.FTE FTE

[Formati del messaggio di log di trasferimento file](#)

[Formati dei messaggi di log di trasferimento file pianificati](#)

[Formato del messaggio di log del monitor MFT](#)

ALW *Esclusione dei tipi di messaggi dal programma di registrazione file autonomo MFT*

Se si desidera escludere un determinato tipo di messaggio dall'output del programma di registrazione file, è possibile utilizzare elementi del tipo di messaggio vuoti.

Esempio

Ad esempio, la seguente definizione del formato arresta i messaggi transferProgress emessi dal programma di registrazione file.

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
  </messageTypes>
</logFormatDefinition>
```

ALW *Definizione di formati personalizzati per il programma di registrazione file autonomo MFT*

È possibile definire un sottoinsieme di tipi di messaggi personalizzati all'interno di una definizione del formato di log per ridurre la quantità di configurazione richiesta per personalizzare il formato del file di log.

Informazioni su questa attività

Se un elemento `messageTypes` non è incluso nel file `FileLoggerFormat.xml`, il formato per quel tipo di messaggio utilizza quello predefinito. È necessario solo specificare i formati che si desidera siano diversi da quelli predefiniti.

Esempio

In questo esempio, la definizione del formato sostituisce il formato predefinito per il tipo di messaggio `transferStarted`, con questa versione ridotta che emette solo l'utente che ha avviato il trasferimento. Tutti gli altri tipi di messaggio utilizzano il formato predefinito poiché non sono inclusi in questa definizione del formato di log:

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
  xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false">/transaction/action/
            @time</insert>
          <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/originator/
            userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

Riferimenti correlati

[Formato di log predefinito del programma di registrazione file autonomo MFT](#)

[Formato del programma di registrazione file autonomo XSD](#)

 *Riduzione dei messaggi duplicati nel programma di registrazione file autonomo MFT*

I messaggi di log duplicati possono verificarsi nel log del programma di registrazione file autonomo. Utilizzando un file `logger.properties` è possibile ottimizzare il programma di registrazione file autonomo e ridurre il numero di duplicati.

Messaggi duplicati nel log del programma di registrazione file

In caso di errore, un messaggio di log potrebbe essere scritto nel log del programma di registrazione file autonomo senza l'utilizzo del messaggio di log da `SYSTEM.FTE/Log#` argomento di cui si sta eseguendo il commit in IBM MQ. Se ciò si verifica, quando il programma di registrazione file autonomo viene riavviato, richiamerà lo stesso messaggio una seconda volta e lo scriverà nuovamente nel file di log. Pianificare la gestione di questi duplicati quando si esaminano i file di log manualmente o quando vengono elaborati automaticamente. Per facilitare il rilevamento dei duplicati, il programma di registrazione file autonomo emette il seguente messaggio nel file di log quando viene avviato:

```
BFGDB0054I: The file logger has successfully started
```

I duplicati si verificano sempre intorno all'ora di inizio del programma di registrazione file autonomo, perché questo è quando viene elaborato l'ultimo messaggio letto prima dell'errore dell'istanza precedente. Sapendo quando la nuova istanza è stata avviata, è possibile rilevare se i duplicati devono essere previsti e se devono essere gestiti o meno.

Riduzione del numero di duplicati

Il programma di registrazione file autonomo raggruppa insieme i messaggi di log che elabora in transazioni per migliorare le prestazioni. Questa dimensione batch è il numero massimo di messaggi duplicati che è possibile visualizzare in caso di errore. Per ridurre il numero di duplicati è possibile ottimizzare la seguente proprietà nel file `logger.properties` :

```
wmqfte.max.transaction.messages
```

Ad esempio, impostandolo su 1, il numero massimo di messaggi duplicati viene ridotto a 1. Tenere presente che la modifica di questo valore ha un effetto sulle prestazioni del programma di registrazione file autonomo, pertanto è necessario un test completo per garantire che ciò non influisca negativamente sul sistema.

Il file `logger.properties` è nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` . Per ulteriori informazioni sulle proprietà disponibili e i relativi effetti, vedere [MFT proprietà di configurazione del programma di registrazione](#)

Configurazione dell'accesso utente per un programma di registrazione file autonomo MFT

In un ambiente di test, è possibile aggiungere qualsiasi nuovo privilegio necessario al proprio account utente normale. In un ambiente di produzione, si consiglia di creare un nuovo utente con le autorizzazioni minime richieste per eseguire il lavoro.

Informazioni su questa attività

È necessario installare il programma di registrazione file autonomo e IBM MQ su un singolo sistema. Configurare le autorizzazioni dell'utente come segue:

Procedura

1. Verificare che l'utente disponga dell'autorizzazione per leggere e, se necessario, eseguire i file installati come parte dell'installazione di Managed File Transfer .
2. Assicurarsi che l'utente disponga dell'autorizzazione per creare e scrivere in qualsiasi file nella directory `logs` che si trova nella directory di configurazione. Questa directory viene utilizzata per un log eventi e, se necessario, per la traccia diagnostica e i file FFDC (First Failure Data Capture).
3. Assicurarsi che l'utente disponga di un proprio gruppo e che non appartenga ad alcun gruppo con ampie autorizzazioni sul gestore code di coordinamento. L'utente non deve essere nel gruppo `mqm`. Su alcune piattaforme, al gruppo `staff` viene automaticamente fornito l'accesso al gestore code; l'utente del programma di registrazione file autonomo non dovrebbe essere nel gruppo `staff`. È possibile visualizzare i record di autorizzazioni per il gestore code stesso e per gli oggetti in esso contenuti utilizzando IBM MQ Explorer. Fare clic con il pulsante destro del mouse sull'oggetto e selezionare **Autorizzazioni oggetto > Gestisci record di autorizzazione**. Nella riga di comando, è possibile utilizzare i comandi `dspmqaaut` (visualizza autorità) o `dmpmqaut` (dump autorità).
4. Utilizzare la finestra **Gestisci record di autorizzazione** nel comando IBM MQ Explorer o `setmqaut` (concessione o revoca dell'autorizzazione) per aggiungere le autorizzazioni per il gruppo dell'utente (su AIX, IBM MQ le autorizzazioni sono associate solo ai gruppi, non ai singoli utenti). Le autorità richieste sono le seguenti:
 - Connettere e richiedere sul gestore code (le librerie di IBM MQ Java richiedono l'autorizzazione di interrogazione per funzionare).
 - Sottoscrivere l'autorizzazione sul SISTEMA `SYSTEM.FTE` .
 - Inserire l'autorizzazione su `SYSTEM.FTE.LOG.RJCT.nome_programma di registrazione` .
 - Ottenere l'autorizzazione su `SYSTEM.FTE.LOG.CMD.nome_programma di registrazione` .

I nomi delle code di comando e di rifiuto forniti sono i nomi predefiniti. Se sono stati scelti nomi di coda differenti quando sono state configurate le code del programma di registrazione file autonomo, aggiungere le autorizzazioni a tali nomi di coda.

Installazione del programma di registrazione database autonomo MFT

Completare la seguente procedura per installare e configurare il programma di registrazione database autonomo.

Informazioni su questa attività

Importante: I logger Managed File Transfer non sono supportati sulla piattaforma IBM i .

Per ulteriori informazioni sul programma di registrazione database autonomo, consultare [“Configurazione di un programma di registrazione MFT”](#) a pagina 807.

Nota: Non è possibile eseguire più di un programma di registrazione database (autonomo o Java EE) sullo stesso schema in un database alla volta. Se si tenta di eseguire questa operazione, si verificherebbe un conflitto durante il tentativo di scrivere i dati del log di trasferimento nel database.

Procedura

1. Installare il proprio software di database utilizzando la relativa documentazione.

Se il supporto JDBC è un componente facoltativo per il database, è necessario installare questo componente.

2. Eseguire il comando **fteCreateLogger** impostando il parametro **-loggerType** su DATABASE per creare un programma di registrazione database autonomo. Per ulteriori informazioni, vedi [fteCreateLogger](#).

Il nome schema predefinito è FTELOG. Se si utilizza un nome schema diverso da FTELOG, è necessario modificare il file SQL fornito appropriato per il database, `ftelog_tables_db2.sql` o `ftelog_tables_oracle.sql`, in modo da riflettere questo nome schema prima di procedere al passo successivo. Per ulteriori informazioni, consultare `wmqfte.database.schema` in [MFT logger configuration properties](#).

3. Creare le tabelle di database richieste utilizzando gli strumenti del database.


Multi Su Multipiattaforme, i file `ftelog_tables_db2.sql` e `ftelog_tables_oracle.sql` contengono comandi SQL che è possibile eseguire per creare le tabelle.

z/OS Su z/OS, il file da eseguire dipende dalla versione di Db2 for z/OS che si sta utilizzando:

- Per Db2 for z/OS 9.0 e versioni precedenti, eseguire il file `ftelog_tables_zos.sql` per creare le tabelle. Questo file crea le tabelle utilizzando un tipo di dati INTEGER per i campi che indicano le dimensioni dei file trasferiti e l'ID tabella associato a ogni trasferimento.
- Per Db2 for z/OS 9.1 e versioni successive, eseguire il file `ftelog_tables_zos_bigint.sql` per creare le tabelle. Questo file crea le tabelle utilizzando un tipo di dati BIGINT per i campi che indicano le dimensioni dei file trasferiti e l'ID tabella associato a ogni trasferimento.

4. Eseguire i comandi MQSC, forniti dal comando **fteCreateLogger**, sul gestore code dei comandi del programma di registrazione per creare le code del programma di registrazione. Il programma di registrazione database autonomo utilizza due code sul gestore code di coordinazione. La prima coda è una coda comandi in cui vengono inseriti i messaggi per controllare il funzionamento del programma di registrazione database autonomo. Il nome predefinito di questa coda comandi è `SYSTEM.FTE.LOG.COMD.nome_logger`. La seconda coda è una coda elementi respinti. Poiché il programma di registrazione database autonomo non elimina mai i messaggi di log, se il programma di registrazione rileva un messaggio che non è in grado di gestire, inserisce il messaggio nella coda di elementi respinti per l'esame e la possibile rielaborazione. Non si consiglia di utilizzare la coda dei messaggi non recapitabili del gestore code per questo scopo, poiché i messaggi rifiutati non hanno un'intestazione DLH e perché i messaggi rifiutati non devono essere combinati con i messaggi inseriti nella coda dei messaggi non recapitabili per altri motivi. Il nome predefinito per la coda di elementi respinti è `SYSTEM.FTE.LOG.RJCT.nome_logger`. Queste due code vengono definite nei file di script MQSC generati dal comando **fteCreateLogger**.

5. [Scegliere un utente e configurare le autorizzazioni](#)

6. Opzionale: È possibile configurare ulteriormente il programma di registrazione database autonomo modificando il file `logger.properties` creato dal comando **fteCreateLogger** nel passo “2” a pagina 817. Questo file è un file delle proprietà Java costituito da coppie chiave - valore. Il file `logger.properties` è nella directory `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name`. Per ulteriori informazioni sulle proprietà disponibili e sui relativi effetti, consultare [MFT proprietà di configurazione del programma di registrazione](#).
7.  **Windows**
Opzionale: Se si sta utilizzando un sistema Windows, è possibile eseguire il programma di registrazione database autonomo come un servizio Windows. Eseguire il comando **fteModifyLogger** con il parametro **-s**. Per ulteriori informazioni, consultare [fteModifyLogger](#).
8. Opzionale: Se il database utilizzato è Oracle o si sta eseguendo la connessione a un database Db2 in remoto, sarà necessario specificare un nome utente e una password che il programma di registrazione utilizzerà per eseguire l'autenticazione con il server di database. Questo nome utente e password vengono specificati in un file di credenziali conforme al formato definito dallo schema `MQMFTCcredentials.xsd`. Per ulteriori informazioni, consultare [Formato file delle credenziali MFT](#). Dopo aver creato il file delle credenziali, è necessario specificare l'ubicazione del file delle credenziali nel file `logger.properties` utilizzando la proprietà `wmqfte.database.credentials.file`.
9. Avviare il programma di registrazione database autonomo utilizzando il comando **fteStartLogger**. Per impostazione predefinita, il programma di registrazione database autonomo viene eseguito in background e il programma di registrazione database autonomo inserisce l'output in un file nella directory `logs`. Se si desidera eseguire il programma di registrazione database autonomo in primo piano e produrre l'output sulla console e sul file di log, aggiungere il parametro **-F** al comando **fteStartLogger**.

Se hai eseguito il passo precedente e hai utilizzato il comando **fteModifyLogger** con il parametro **-s** su Windows, il programma di registrazione database autonomo viene avviato come un servizio Windows.

Attività correlate

[“Configurazione dell'accesso utente per un programma di registrazione database autonomo MFT” a pagina 819](#)

In un ambiente di test, è possibile aggiungere qualsiasi nuovo privilegio necessario al proprio account utente normale. In un ambiente di produzione, si consiglia di creare un nuovo utente con le autorizzazioni minime richieste per eseguire il lavoro.

Riferimenti correlati

[Proprietà di configurazione del programma di registrazione MFT](#)

[Logger fteStart](#)

[Programma di registrazione fteModify](#)

[Autorità per il logger MFT](#)

Utilizzo di MFT con un database remoto

È possibile utilizzare il programma di registrazione Managed File Transfer per comunicare con un database su un sistema remoto.

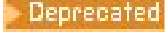
Informazioni su questa attività

Se si dispone di un database installato su una macchina diversa da quella su cui è installato Managed File Transfer, completare la seguente procedura. La procedura si applica sia a Db2 che a Oracle, se non diversamente specificato.

Procedura

1. Installare un client database sul sistema su cui è stato installato Managed File Transfer.

2. Aggiungere il server database remoto alla configurazione del client database locale. Questo aggiornamento della configurazione è necessario per Managed File Transfer e IBM MQ per accedere correttamente al database.
3. Specificare le nuove proprietà nel file di `logger.properties` per connettersi al database utilizzando il file delle credenziali `wmfte.database.credentials.file`.

Nota:  Le versioni precedenti di Managed File Transfer utilizzavano le proprietà `wmqfte.oracle.user` o `wmqfte.database.user`, `wmqfte.oracle.password` o `wmqfte.database.password`. Queste proprietà sono ora obsolete. Utilizzare invece `wmfte.database.credentials.file`.

4. **SoloOracle** : per consentire una connessione remota al database, modificare la stanza XAResourceManager nel file `qm.ini` del gestore code di coordinamento nel seguente modo (assicurandosi di modificare il nome del database, il nome utente e la password utente in modo che corrispondano alle proprie informazioni):

```
Oracle_XA+Acc=P/ftelog/  
qgw783jhT+SesTm=35+DB=FTEAUDIT1+SqlNet=FTEAUDIT1+threads=false,
```

la modifica viene evidenziata in grassetto.
5. **SoloOracle** : specificare host e porta nel file `logger.properties`, utilizzando le proprietà `wmqfte.oracle.host` e `wmqfte.oracle.port`. I valori predefiniti per l'host e la porta consentono di lavorare con un client di database locale, quindi se è stato precedentemente utilizzato un database locale, è possibile che questi valori non siano stati impostati.

Riferimenti correlati

[Proprietà di configurazione del programma di registrazione MFT](#)

Configurazione dell'accesso utente per un programma di registrazione database autonomo MFT

In un ambiente di test, è possibile aggiungere qualsiasi nuovo privilegio necessario al proprio account utente normale. In un ambiente di produzione, si consiglia di creare un nuovo utente con le autorizzazioni minime richieste per eseguire il lavoro.

Informazioni su questa attività

Il numero e il tipo di account utente necessari per eseguire il programma di registrazione database autonomo dipendono dal numero di sistemi utilizzati. È possibile installare il programma di registrazione database autonomo, IBM MQ e il proprio database su un singolo sistema o su due sistemi. Il programma di registrazione database autonomo deve trovarsi sullo stesso sistema di IBM MQ. I componenti possono essere installati nelle seguenti topologie:

Logger del database autonomo, IBM MQ e il database tutti sullo stesso sistema

È possibile definire un singolo utente del sistema operativo da utilizzare con tutti e tre i componenti. Questa è una configurazione adatta per il programma di registrazione database autonomo. Il programma di registrazione database autonomo utilizza la modalità Bindings per connettersi a IBM MQ e una connessione nativa per connettersi al database.

Programma di registrazione del database autonomo e IBM MQ su un sistema, il database su un sistema separato

Si creano due utenti per questa configurazione: un utente del sistema operativo sul sistema che esegue il programma di registrazione database autonomo e un utente del sistema operativo con accesso remoto al database sul server database. Questa è una configurazione adatta per il programma di registrazione database autonomo che utilizza un database remoto. Il programma di registrazione database autonomo utilizza la modalità ... Bind per connettersi a IBM MQ e una connessione client per accedere al database.

Come esempio, il resto di queste istruzioni presuppone che l'utente sia denominato `ftelog`, ma è possibile utilizzare qualsiasi nome utente. Configurare le autorizzazioni dell'utente come segue:

Procedura

1. Verificare che l'utente disponga dell'autorizzazione per leggere e, se necessario, eseguire i file installati come parte dell'installazione di Managed File Transfer Remote Tools and Documentation.
2. Verificare che l'utente disponga dell'autorizzazione per creare e scrivere in qualsiasi file nella directory logs (nella directory di configurazione). Questa directory viene utilizzata per un log eventi e, se necessario, per la traccia diagnostica e i file FFDC.
3. Verificare che l'utente disponga di un proprio gruppo e che non sia presente anche in gruppi con autorizzazioni di ampio respiro sul gestore code di coordinamento. L'utente non deve essere nel gruppo mqm. Su alcune piattaforme, al gruppo di staff viene automaticamente fornito l'accesso al gestore code; l'utente del programma di registrazione database autonomo non deve far parte del gruppo di staff. È possibile visualizzare i record di autorizzazioni per il gestore code stesso e per gli oggetti in esso contenuti utilizzando IBM MQ Explorer. Fare clic con il pulsante destro del mouse sull'oggetto e selezionare **Autorizzazioni oggetto > Gestisci record di autorizzazione**. Nella riga di comando, è possibile utilizzare i comandi `dspmqaut` (visualizza autorità) o `dmpmqaut` (dump autorità).
4. Utilizzare la finestra **Gestisci record di autorizzazione** nel comando IBM MQ Explorer o `setmqaut` (concessione o revoca dell'autorizzazione) per aggiungere le autorizzazioni per il gruppo dell'utente (su AIX, IBM MQ le autorizzazioni sono associate solo ai gruppi, non ai singoli utenti). Le autorità richieste sono le seguenti:

- Connettere e richiedere sul gestore code (le librerie di IBM MQ Java richiedono l'autorizzazione di interrogazione per funzionare).
- Sottoscrivere l'autorizzazione sul SISTEMA SYSTEM.FTE .
- Inserire l'autorizzazione su SYSTEM.FTE.LOG.RJCT.*nome_programma di registrazione* .
- Ottenere l'autorizzazione su SYSTEM.FTE.LOG.CMD.*nome_programma di registrazione* .

I nomi delle code di comando e di rifiuto forniti sono i nomi predefiniti. Se sono stati scelti nomi di coda differenti quando sono state configurate le code del programma di registrazione database autonomo, aggiungere le autorizzazioni a tali nomi di coda.

5. Eseguire la configurazione utente specifica per il database che si sta utilizzando.

- Se il database è Db2, effettuare le seguenti operazioni:

Esistono diversi meccanismi per la gestione degli utenti del database con Db2. Queste istruzioni si applicano allo schema predefinito in base agli utenti del sistema operativo.

- Assicurarsi che l'utente `fte1og` non si trovi in alcun gruppo di gestione Db2 (ad esempio, `db2iadm1`, `db2fadm1` o `dasadm1`)
- Fornire all'utente l'autorizzazione per connettersi al database e l'autorizzazione per selezionare, inserire e aggiornare le tabelle create come parte del [Passo 2: creare le tabelle di database richieste](#)

- Se il database è Oracle, effettuare le seguenti operazioni:

- Assicurarsi che l'utente `fte1og` non si trovi in alcun gruppo di gestione Oracle (ad esempio, `ora_dba` su Windows o `dba` su AIX and Linux)
- Fornire all'utente l'autorizzazione per connettersi al database e l'autorizzazione per selezionare, inserire e aggiornare le tabelle create come parte del [Passo 2: creare le tabelle di database richieste](#)

Configurazioni alternative per un logger autonomo MFT

Generalmente, un programma di registrazione autonomo Managed File Transfer, sia esso un file o un tipo di database, si trova sullo stesso sistema del gestore code di coordinamento ed è connesso al gestore code di coordinazione in modalità di bind IBM MQ. Tuttavia, può anche essere installato sullo stesso sistema di qualsiasi gestore code che abbia la connettività al gestore code di coordinamento. Il programma di registrazione autonomo riceve i messaggi utilizzando una sottoscrizione, che il programma di registrazione autonomo crea automaticamente. Questa è la configurazione descritta nelle istruzioni di installazione.

Tuttavia, se si hanno considerazioni specifiche del sito, è possibile configurare un programma di registrazione autonomo per ricevere i messaggi in altri due modi, controllati dalla proprietà `wmqfte.message.source.type`. Questa proprietà è descritta in [MFT proprietà di configurazione del programma di registrazione](#).

Sottoscrizione amministrativa

Per impostazione predefinita, un programma di registrazione autonomo crea la propria sottoscrizione a `SYSTEM.FTE/Log/#`, utilizzando le opzioni di sottoscrizione durevole predefinite e una sottoscrizione gestita (ossia, il gestore code controlla la coda di backup utilizzata per conservare i messaggi prima che vengano trasmessi all'applicazione). Se sono richieste altre opzioni sulla sottoscrizione o sulla coda, è possibile creare una sottoscrizione da soli, impostare le opzioni richieste e configurare il programma di registrazione autonomo per utilizzare tale sottoscrizione. Ricordarsi di aggiungere l'autorizzazione per il programma di registrazione autonomo per utilizzare la sottoscrizione creata.

Un esempio di utilizzo di questa configurazione è la partizione dello spazio di log utilizzando due sottoscrizioni jolly, per inviare i log dagli agent il cui nome inizia con `FINANCE` in un database e i log dagli agent che iniziano con `ACCOUNTING` in un altro. Questo tipo di configurazione richiede due istanze del programma di registrazione autonomo, ciascuna con il proprio file `logger.properties` che fa riferimento alla richiesta e la propria coda comandi e coda di elementi respinti.

Per raccogliere i messaggi di log solo dagli agent i cui nomi iniziano con `ACCOUNTING`, creare un oggetto sottoscrizione sul gestore code di coordinazione con una stringa di argomenti `SYSTEM.FTE/Log/ACCOUNTING*`. Impostare il valore **Utilizzo carattere jolly** su **Carattere jolly di livello carattere**. È inoltre necessario aggiungere voci al file `logger.properties` per il logger. Ad esempio, se si crea un oggetto sottoscrizione denominato `ACCOUNTING.LOGS` con queste impostazioni, aggiungere le voci seguenti al file `logger.properties`:

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

Il programma di registrazione autonomo gestisce i messaggi di log che iniziano con la stringa di argomenti `SYSTEM.FTE/Log/`. È possibile specificare una stringa di argomenti più restrittiva, ma non è possibile specificare una stringa meno restrittiva. Se si specifica una stringa meno restrittiva in errore, tutte le pubblicazioni relative a una stringa di argomenti diversa da `SYSTEM.FTE/Log/` passano alla coda di elementi respinti e il programma di registrazione autonomo produce il messaggio di errore `BFGDB0002E`. Questo messaggio di errore implica un problema con la configurazione del programma di registrazione autonomo.

Coda

La topologia tipica è quella in cui il programma di registrazione autonomo viene eseguito sullo stesso sistema del gestore code di coordinamento. Se ciò non è possibile, è possibile creare una sottoscrizione sul gestore code di coordinamento utilizzando una coda su un altro gestore code come destinazione della sottoscrizione (utilizzando una definizione della coda remota o utilizzando la proprietà `DESTQMGR` della sottoscrizione). Il programma di registrazione può quindi essere eseguito sul sistema che ospita il secondo gestore code e leggere i messaggi dalla coda. Per garantire l'integrità delle transazioni, il programma di registrazione autonomo deve sempre connettersi al proprio gestore code in modalità di `bind`. È necessario definire la coda di elementi respinti e la coda comandi sullo stesso gestore code a cui si connette il programma di registrazione autonomo. I gestori code devono essere IBM WebSphere MQ 7.5 o successivi.

Ad esempio, per raccogliere i messaggi di log che vengono inseriti nella coda `USER.QUEUE` da una richiesta, aggiungere le seguenti voci al file `logger.properties`:

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

Installazione del programma di registrazione database Java EE per MFT

Seguire queste istruzioni per installare e configurare il programma di registrazione database JEE da utilizzare con Managed File Transfer.

Informazioni su questa attività

Per ulteriori informazioni sul programma di registrazione database Java EE , consultare l'argomento [“Configurazione di un programma di registrazione MFT” a pagina 807.](#)

Nota: Non è possibile eseguire un programma di registrazione database Java EE contemporaneamente a un programma di registrazione autonomo, a meno che tali programmi di registrazione non utilizzino istanze separate del database.

Procedura

1. Prima di installare il programma di registrazione database Java EE , è necessario preparare il proprio ambiente. Utilizzare le istruzioni nell'argomento [“Preparazione all'installazione del programma di log del database Java EE per MFT” a pagina 822.](#)
2. Installare il programma di registrazione database Java EE in un server delle applicazioni conforme Java Platform, Enterprise Edition (Java EE) o Jakarta EE .

Per istruzioni, consultare [“Installazione del programma di registrazione database Java EE per MFT con WebSphere Application Server traditional 9.0” a pagina 825](#)

Attività correlate

[“Preparazione all'installazione del programma di log del database Java EE per MFT” a pagina 822](#)

Seguire queste istruzioni per preparare l'ambiente Managed File Transfer prima di installare il programma di registrazione database Java EE .

[“Installazione del programma di registrazione database Java EE per MFT con WebSphere Application Server traditional 9.0” a pagina 825](#)

Seguire queste istruzioni per installare e configurare il programma di registrazione database Java Platform, Enterprise Edition (Java EE) per Managed File Transfer con WebSphere Application Server traditional 9.0.

[“Configurazione dell'accesso utente per il logger del database Java EE per MFT” a pagina 830](#)

Quando si configura il programma di registrazione database Java Platform, Enterprise Edition (Java EE) per Managed File Transfer, sono necessari gli account utente per accedere a IBM MQ, al database e al sistema operativo. Il numero di utenti del sistema operativo richiesto dipende dal numero di sistemi utilizzati per ospitare questi componenti.

[“Migrazione dal programma di registrazione database autonomo al programma di registrazione database di Java EE per MFT” a pagina 831](#)

È possibile eseguire la migrazione dal programma di registrazione database autonomo al programma di registrazione database Java EE . È necessario arrestare il programma di registrazione database autonomo e installare il programma di registrazione database JEE. Per evitare la perdita o la duplicazione delle voci di log, è necessario arrestare la pubblicazione dei messaggi nel sistema SYSTEM.FTE prima di arrestare il programma di registrazione database autonomo e riavviarlo dopo aver installato il programma di registrazione database Java EE . Eseguire il backup del database prima della migrazione.

Riferimenti correlati

[Autorità per il logger MFT](#)

Preparazione all'installazione del programma di log del database Java EE per MFT

Seguire queste istruzioni per preparare l'ambiente Managed File Transfer prima di installare il programma di registrazione database Java EE .

Informazioni su questa attività

Per ulteriori informazioni sul programma di registrazione database Java EE , consultare l'argomento [“Configurazione di un programma di registrazione MFT” a pagina 807.](#)

Procedura

1. Installare il proprio software di database utilizzando la relativa documentazione.

Se il supporto JDBC è un componente facoltativo per il database, è necessario installare questo componente.

2. Creare un database utilizzando gli strumenti forniti dal proprio database. Il database deve avere un tablespace e una dimensione pagina bufferpool di almeno 8K.

Il nome schema predefinito è FTELOG. Se si utilizza un nome schema diverso da FTELOG, è necessario modificare il file SQL fornito appropriato per il proprio database, `ftelog_tables_db2.sql` o `ftelog_tables_oracle.sql`, per rispecchiarlo prima di procedere al passo successivo.

Nota: I file `ftelog_tables_db2.sql` e `ftelog_tables_oracle.sql` si trovano nel percorso file `<MQ-installation-path>/mqft/sql`

3. Creare le tabelle di database richieste utilizzando gli strumenti del database.

Multi Su Multipiattaforme, i file `ftelog_tables_db2.sql` e `ftelog_tables_oracle.sql` contengono comandi SQL che è possibile eseguire per creare le tabelle.

z/OS Su z/OS, il file da eseguire dipende dalla versione di Db2 for z/OS che si sta utilizzando:

- Per Db2 for z/OS 9.0 e versioni precedenti, eseguire il file `ftelog_tables_zos.sql` per creare le tabelle. Questo file crea le tabelle utilizzando un tipo di dati INTEGER per i campi che indicano le dimensioni dei file trasferiti e l'ID tabella associato a ogni trasferimento.
- Per Db2 for z/OS 9.1 e versioni successive, eseguire il file `ftelog_tables_zos_bigint.sql` per creare le tabelle. Questo file crea le tabelle utilizzando un tipo di dati BIGINT per i campi che indicano le dimensioni dei file trasferiti e l'ID tabella associato a ogni trasferimento.

4. Se è stato modificato il nome dello schema da FTELOG, è necessario modificare il nome dello schema nel file EAR. Per ulteriori informazioni, fare riferimento a [“Modifica del nome dello schema nel programma di registrazione database Java EE per MFT”](#) a pagina 823.

5. Creare una coda di elementi respinti in IBM MQ.

Poiché il programma di registrazione non elimina mai i messaggi di log, se il programma di registrazione rileva un messaggio che non può gestire, inserisce il messaggio nella coda di elementi respinti per l'esame e la possibile rielaborazione. Non utilizzare la coda di messaggi non recapitabili del gestore code per questo scopo, perché i messaggi rifiutati non dispongono di un'intestazione DLH e perché i messaggi rifiutati non devono essere combinati con i messaggi inseriti nella coda di messaggi non recapitabili per altri motivi. Il comando **fteCreateLogger** crea una coda di elementi respinti. Il nome predefinito per questa coda elementi respinti è `SYSTEM.FTE.LOG.RJCT.nome_programma_di_registrazione`

6. Seguire le istruzioni riportate nell'argomento [“Configurazione dell'accesso utente per il logger del database Java EE per MFT”](#) a pagina 830.

Operazioni successive

Installare il programma di registrazione del database Java EE in un server delle applicazioni compatibile con Java EE o Jakarta EE . Utilizzare le istruzioni in [“Installazione del programma di registrazione database Java EE per MFT con WebSphere Application Server traditional 9.0”](#) a pagina 825

Modifica del nome dello schema nel programma di registrazione database Java EE per MFT

Il programma di registrazione database Java Platform, Enterprise Edition (Java EE) può utilizzare un database che ha un nome schema non predefinito. È necessario modificare il nome schema nel file EAR del programma di registrazione database Java EE .

Informazioni su questa attività

Per modificare il nome dello schema utilizzato dal programma di registrazione database Java EE , completare la seguente procedura:

Procedura

1. Estrarre il file JAR JPA dal file EAR utilizzando il seguente comando:

```
jar -xvf ear_file lib/jpa_file
```

dove:

- *ear_file* è com.ibm.wmqfte.databaselogger.jee.oracle.ear o com.ibm.wmqfte.databaselogger.jee.ear a seconda che si stia utilizzando Db2 o Oracle.
- *jpa_file* è com.ibm.wmqfte.web.jpa.oracle.jar o com.ibm.wmqfte.web.jpa.jar a seconda che si stia utilizzando Db2 o Oracle.

2. Estrarre il file persistence.xml dal file JAR JPA utilizzando il seguente comando:

```
jar -xvf lib/jpa_file META_INF/persistence.xml
```

dove:

- *jpa_file* è com.ibm.wmqfte.web.jpa.oracle.jar o com.ibm.wmqfte.web.jpa.jar a seconda che si stia utilizzando Db2 o Oracle.

3. Modificare il file persistence.xml per modificare la seguente riga:

```
<property name="openjpa.jdbc.Schema" value="schema_name"/>
```

dove

- *schema_name* è il nome schema che si desidera utilizzare.

4. Aggiornare il JAR JPA con il file persistence.xml modificato utilizzando il seguente comando:

```
jar -uvf lib/jpa_file META_INF/persistence.xml
```

dove:

- *jpa_file* è com.ibm.wmqfte.web.jpa.oracle.jar o com.ibm.wmqfte.web.jpa.jar a seconda che si stia utilizzando Db2 o Oracle.

5. Aggiornare il file EAR con il file JAR JPA modificato utilizzando il seguente comando:

```
jar -uvf ear_file lib/jpa_file
```

dove:

- *ear_file* è com.ibm.wmqfte.databaselogger.jee.oracle.ear o com.ibm.wmqfte.databaselogger.jee.ear a seconda che si stia utilizzando Db2 o Oracle.
- *jpa_file* è com.ibm.wmqfte.web.jpa.oracle.jar o com.ibm.wmqfte.web.jpa.jar a seconda che si stia utilizzando Db2 o Oracle.

Operazioni successive

Utilizzare il file EAR modificato per installare il programma di registrazione database Java EE .

Attività correlate

[“Installazione del programma di registrazione database Java EE per MFT con WebSphere Application Server traditional 9.0” a pagina 825](#)

Seguire queste istruzioni per installare e configurare il programma di registrazione database Java Platform, Enterprise Edition (Java EE) per Managed File Transfer con WebSphere Application Server traditional 9.0.

Impostazione del percorso della libreria nativa in WebSphere Application Server traditional 9.0

Se si distribuisce l'applicazione del programma di registrazione database Java Platform, Enterprise Edition (Java EE) su WebSphere Application Server traditional 9.0 e si desidera utilizzare le connessioni in modalità bind tra l'applicazione e IBM MQ, è necessario configurare il fornitore di messaggistica IBM MQ con l'ubicazione delle librerie native IBM MQ sul sistema.

Informazioni su questa attività

Se non si imposta il percorso della libreria nativa nel server delle applicazioni, si potrebbe ricevere il seguente messaggio di errore nel log di uscita del sistema WebSphere Application Server traditional 9.0 :

```
A connection could not be made to WebSphere MQ for the following reason:  
CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnf' was not found. [3=mqjbnf]
```

Utilizzare la console di amministrazione WebSphere Application Server traditional 9.0 per completare la seguente procedura:

Procedura

1. Nel pannello di navigazione, espandere **Risorse > JMS > Provider JMS**.
2. Selezionare il provider di messaggistica IBM MQ che si trova nell'ambito corretto per la factory di connessione o la specifica di attivazione che crea la connessione in modalità bind.
Nota: Le informazioni sul percorso nativo nell'ambito `Server` vengono utilizzate in preferenza alle informazioni sul percorso nativo negli ambiti superiori e le informazioni sul percorso nativo nell'ambito `Node` vengono utilizzate in preferenza alle informazioni sul percorso nativo nell'ambito `Cell`.
3. In Proprietà generali, nel campo **Percorso libreria nativa**, immettere il nome completo della directory che contiene le librerie native IBM MQ.
Ad esempio, su Linux immettere `/opt/mqm/java/lib`. Immettere un solo nome di directory.
4. Fare clic su **OK**.
Una volta impostato il percorso, è necessario salvare le modifiche alla configurazione principale per rendere effettive le modifiche.
5. Riavviare il server delle applicazioni per aggiornare la configurazione.
6. Richiesto: Riavviare il server delle applicazioni una seconda volta per caricare le librerie.

Attività correlate

[“Installazione del programma di registrazione database Java EE per MFT con WebSphere Application Server traditional 9.0” a pagina 825](#)

Seguire queste istruzioni per installare e configurare il programma di registrazione database Java Platform, Enterprise Edition (Java EE) per Managed File Transfer con WebSphere Application Server traditional 9.0.

Installazione del programma di registrazione database Java EE per MFT con WebSphere Application Server traditional 9.0

Seguire queste istruzioni per installare e configurare il programma di registrazione database Java Platform, Enterprise Edition (Java EE) per Managed File Transfer con WebSphere Application Server traditional 9.0.


Prima di iniziare

Prima di installare l'applicazione programma di registrazione database JEE, seguire le istruzioni riportate negli argomenti [“Preparazione all'installazione del programma di log del database Java EE per MFT” a pagina 822](#) e [“Impostazione del percorso della libreria nativa in WebSphere Application Server traditional 9.0” a pagina 825](#).

Informazioni su questa attività

Per ulteriori informazioni sul programma di registrazione database Java EE , consultare [“Configurazione di un programma di registrazione MFT”](#) a pagina 807.

Procedura

1. Impostare il provider JDBC XA:
 - a) Seleziona **Resources > JDBC > JDBC Providers** dalla navigazione della console di amministrazione WebSphere Application Server traditional 9.0 .
 - b) Creare un provider JDBC utilizzando la procedura guidata della console, facendo clic su **Nuovo**.
 - c) Nel passo 1 della procedura guidata, selezionare il database che si sta utilizzando dall'elenco **Tipo di database** e il tipo di fornitore associato dall'elenco **Tipo di fornitore** . Dall'elenco **Tipo di implementazione** , selezionare **Origine dati XA**. Fare clic su **Avanti**.
 È possibile rimuovere un riferimento a db2jcc_license_cisuz.jar e modificare db2jcc.jar in db2jcc4.jar, ossia la versione del file jar fornito con la versione più recente di Db2o la versione locale.
 - d) Al passo 2 della procedura guidata, verificare che l'ubicazione della directory dei file jar del database richiesti sia impostata correttamente. Fare clic su **Avanti**.
 - e) Fare clic su **Fine** nella pagina di riepilogo per creare il provider JDBC .
2. Creare alias di autenticazione. Si crea un alias per l'origine dati e un altro per IBM MQ:
 - a) Seleziona **Sicurezza > Sicurezza globale** dalla navigazione della console di gestione WebSphere Application Server traditional 9.0 .
 - b) Nell'intestazione **Autenticazione** , espandere **JAAS (Java Authentication and Authorization Service)**.
 - c) Fare clic su **J2C**. Viene aperta la pagina dell'alias di autenticazione.
 - d) Creare un alias di autenticazione per l'origine dati:
 - i) Fare clic su **Nuovo**.
 - ii) Immettere i dettagli per **Alias, ID utente, Passworde Descrizione**. I dettagli immessi nei campi **User ID** e **Password** devono corrispondere ai dettagli immessi quando è stato creato l'utente del database. Per ulteriori informazioni, fare riferimento a [“Configurazione dell'accesso utente per il logger del database Java EE per MFT”](#) a pagina 830.
 - iii) Fare clic su **OK**.
 - e) Creare un alias di autenticazione per IBM MQ:
 - i) Fare clic su **Nuovo**.
 - ii) Immettere i dettagli per **Alias, ID utente, Passworde Descrizione**. i dettagli immessi nei campi **ID utente** e **Password** devono corrispondere alle impostazioni utente e password per l'installazione di IBM MQ .
 - iii) Fare clic su **OK**.
3. Creare un'origine dati:
 - a) Selezionare **Risorse > JDBC > Origini dati** dalla navigazione della console di gestione WebSphere Application Server traditional 9.0 .
 - b) Selezionare l'elenco a discesa **Ambito** e modificare l'ambito nel valore appropriato. Ad esempio, `Node=yourNode`, `Server=yourServer`.
 - c) Creare un'origine dati utilizzando la procedura guidata della console, facendo clic su **Nuovo**.
 - d) Al passo 1 della procedura guidata, nel campo **Nome origine dati** , immettere `wmqfte-database` e nel campo **Nome JNDI** , immettere `jdbc/wmqfte-database`. Fare clic su **Avanti**.
 - e) Al passo 2 della procedura guidata, utilizzare l'elenco a discesa **Seleziona un provider JDBC** esistente per selezionare il provider JDBC creato nei passi precedenti. Fare clic su **Avanti**.
 - f) **Db2:** Al passo 3 della procedura guidata, immettere 4nel campo **Tipo di driver** .

- g) **Db2:** immettere i dettagli nei campi **Nome database**, **Nome server** **Numero porta** e fare clic su **Avanti**.
- Oracle:** immettere l'URL di connessione nel campo **URL** e scegliere l'helper di archivio dati corretto nel campo **Nome classe helper archivio dati** .
- Oracle RAC:** quando ci si connette a un Oracle Real Application Cluster, l'URL di connessione deve includere le informazioni host necessarie per la connessione a tutte le istanze disponibili del database.
- h) Al passo 4 della procedura guidata, selezionare il nome dell'alias di autenticazione dell'origine dati definito al passo 2d dall'elenco **Alias di autenticazione per il ripristino XA** . Selezionare lo stesso nome dagli elenchi **Alias di autenticazione gestito dal componente** e **Alias di autenticazione gestito dal contenitore** .
- i) Fare clic su **Fine** nella pagina di riepilogo per creare l'origine dati.
4. Opzionale: Verificare la configurazione dell'origine dati:
- a) Selezionare **Risorse > JDBC > Origini dati** dalla navigazione della console di gestione WebSphere Application Server traditional 9.0 .
- b) Fare clic su **Verifica connessione** .
5. Crea un argomento.
- a) Dalla navigazione nella console di amministrazione WebSphere Application Server traditional 9.0 , fare clic su **Risorse > JMS > Argomenti**.
- b) Selezionare l'elenco a discesa **Ambito** e modificare l'ambito nel valore appropriato. Ad esempio, `Node=yourNode` , `Server=yourServer`.
- c) Fare clic su **Nuovo**.
- d) Fare clic sul provider di messaggistica **IBM MQ**.
- e) Sul pannello **Amministrazione** della pagina delle proprietà per l'argomento, scegliere valori univoci per i campi **Nome** e **Nome JNDI** , a cui si farà riferimento successivamente nella configurazione.
- f) Nel pannello **IBM MQ** , immettere `SYSTEM.FTE/Log/#` nel campo **Nome argomento** .
6. Creare una specifica di attivazione:
- a) Dalla navigazione della console di gestione WebSphere Application Server traditional 9.0 , fare clic su **Risorse > JMS > Specifiche di attivazione**.
- b) Selezionare l'elenco a discesa **Ambito** e modificare l'ambito nel valore appropriato. Ad esempio, `Node=yourNode` , `Server=yourServer`.
- c) Fare clic su **Nuovo**.
- d) Fare clic sul provider di messaggistica **IBM MQ**.
- e) Nel passo 1 della procedura guidata, scegliere valori univoci per i campi **Nome** e **Nome JNDI** , a cui si farà nuovamente riferimento in seguito nella configurazione.
- f) Nel passo 1.1, immettere il nome JNDI per l'argomento impostato al punto 5 nel campo **Nome JNDI destinazione** .
- g) Dall'elenco **Tipo di destinazione** , selezionare **Argomento**.
- h) Nel passo 1.2 della procedura guidata, selezionare **Sottoscrizione durevole**. Immettere `SYSTEM.FTE.DATABASELOGGER.AUTO` nel campo **Nome sottoscrizione** .
- i) Nel passo 2 della procedura guidata, selezionare **Immettere tutte le informazioni richieste in questa procedura guidata**.
- j) Nel passo 2.1, immettere il nome del gestore code nel campo **Nome del gestore code o del gruppo di condivisione code** .
- k) Nel passo 2.2, selezionare il metodo di trasporto scelto dall'elenco **Trasporto** . Se si seleziona **Bind**, non sono richieste altre informazioni. Se si seleziona **Client** o **Bind, quindi client**, immettere i dettagli per **Nome host**, **Portae Canale di connessione server**.

- l) Opzionale: Fare clic su **Verifica connessione** per verificare che il gestore code sia presente. Tuttavia, è possibile ricevere NOT_AUTHORIZED fino a quando non si fa riferimento all'alias di autenticazione nel passo 6n.
 - m) Fare clic su **Salva**.
 - n) Fare clic sul nome della specifica di attivazione creata. Nella sezione **Proprietà generali** della scheda **Configurazione**, scorrere fino al pannello **Avanzate** e immettere un nome univoco per identificare la connessione IBM MQ nel campo **ID client**. È necessario completare questo passo o la propria connessione viene rifiutata da IBM MQ con il codice di errore JM5CC0101.
 - o) Se si sceglie **Client** come metodo di trasporto, scorrere fino al pannello **Impostazioni di sicurezza** e selezionare l'alias di autenticazione definito al passo 8 dall'elenco **Alias di autenticazione**.
 - p) Fare clic su **Applica**.
 - q) Nella sezione **Ulteriori proprietà** della scheda **Configurazione**, fare clic su **Proprietà avanzate**. Nella sezione **Consumatore connessione** del pannello **Proprietà avanzate**, immettere 1 nel campo **Numero massimo di sessioni server**.
- Nota:** Assicurarsi di completare questa operazione prima di proseguire. In caso contrario, il programma di registrazione potrebbe non funzionare correttamente.
- r) Nella sezione **Ulteriori proprietà** della scheda **Configurazione**, fare clic su **Proprietà avanzate**. Impostare il valore di **Arresta endpoint se la consegna del messaggio non riesce** su un minimo di 1.

Se il valore della proprietà **_numberOfFailedAttemptsBeforeReject** è impostato su più di 1 (per ulteriori informazioni, vedere 9j), impostare **Arresta endpoint se la consegna del messaggio non riesce** almeno sul valore della proprietà **_numberOfFailedAttemptsBeforeReject**. Ciò impedisce all'endpoint di arrestarsi quando viene ricevuto un messaggio che non può essere elaborato (ad esempio, un messaggio di log di trasferimento non corretto). Per ulteriori informazioni, consultare [MFT Gestione e rifiuto degli errori del programma di registrazione](#).

7. Creare un factory di connessione code.

- a) Dalla navigazione della console di gestione WebSphere Application Server traditional 9.0, fare clic su **Risorse > JMS > Factory di connessione code**.
- b) Selezionare l'elenco a discesa **Ambito** e modificare l'ambito nel valore appropriato. Ad esempio, `Node=yourNode`, `Server=yourServer`.
- c) Fare clic su **Nuovo**.
- d) Fare clic sul provider di messaggistica **IBM MQ**.
- e) Nel passo 1 della procedura guidata, scegliere valori univoci per i campi **Nome** e **Nome JNDI**, a cui si farà nuovamente riferimento in seguito nella configurazione.
- f) Nel passo 2, selezionare **Immettere tutte le informazioni richieste in questa procedura guidata**.
- g) Nel passo 2.1, immettere il nome del gestore code nel campo **Nome del gestore code o del gruppo di condivisione code**.
- h) Nel passo 2.2, selezionare il metodo di trasporto scelto dall'elenco **Trasporto**. Se si seleziona **Bind**, non sono richieste altre informazioni. Se si seleziona **Client** o **Bind, quindi client**, immettere i dettagli per **Nome host**, **Porta** e **Canale di connessione server**.
- i) Opzionale: Fare clic su **Verifica connessione** per verificare che il gestore code sia presente. Tuttavia, è possibile ricevere NOT_AUTHORIZED fino a quando non si fa riferimento all'alias di autenticazione nel passaggio 7h.
- j) Se è stato selezionato **Client** o **Bind e client** come metodo di trasporto, fare clic sul nome della factory di connessione code appena creata. Scorrere verso il basso il pannello **Impostazioni di protezione** della scheda **Configurazione** e selezionare l'alias di autenticazione definito nel passo 2e dagli elenchi **Alias di autenticazione per il ripristino XA** e **Alias di autenticazione gestito da contenitore**.

8. Creare una coda elementi respinti in WebSphere Application Server:

- a) Dalla navigazione della console di gestione WebSphere Application Server traditional 9.0 , fare clic su **Risorse > JMS > Code**.
 - b) Selezionare l'elenco a discesa **Ambito** e modificare l'ambito nel valore appropriato. Ad esempio, Node=yourNode, Server=yourServer.
 - c) Fare clic su **Nuovo**.
 - d) Fare clic sul provider di messaggistica **IBM MQ**.
 - e) Scegliere valori univoci per i campi **Nome** e **Nome JNDI** , a cui si farà nuovamente riferimento successivamente nella configurazione.
 - f) Immettere SYSTEM.FTE.LOG.RJCT.*logger_name* nel campo **Nome coda** . Assicurarsi di aver creato questa coda sul gestore code di coordinamento.
 - g) Immettere il proprio nome gestore code nel campo **Nome gestore code** .
 - h) Fare clic su **OK**.
9. Installare l'applicazione del programma di registrazione database JEE:
- a) Dalla console di gestione WebSphere Application Server traditional 9.0 , selezionare **Applicazioni > Nuova applicazione**.
 - b) Selezionare l'elenco a discesa **Ambito** e modificare l'ambito nel valore appropriato. Ad esempio, Node=yourNode, Server=yourServer.
 - c) Dall'elenco di opzioni, selezionare **Nuova applicazione enterprise**.
 - d) Nella pagina **Preparazione per l'installazione dell'applicazione** , selezionare il file com.ibm.wmqfte.databaselogger.jee.ear o il file com.ibm.wmqfte.databaselogger.jee.oracle.ear dalla directory *MQ_INSTALLATION_PATH/mqft/web* dell'installazione di Managed File Transfer Service e fare clic su **Avanti**.
 - e) Nella seguente schermata, selezionare **Dettagliato** per mostrare tutte le opzioni di installazione e i parametri e fare clic su **Avanti**.
 - f) Fare clic su **Avanti** attraverso i passi 1-4 della procedura guidata per accettare i valori predefiniti.
 - g) Nel passo 5 della procedura guidata, **Collegare i listener per i bean basati sui messaggi**, scorrere fino alla sezione **Collegamenti listener** . Fare clic su **Specifica di attivazione**.
Immettere i valori richiesti per i seguenti campi:
Nome JNDI della risorsa di destinazione
Il nome JNDI specificato durante la creazione di una specifica di attivazione nel passo 6d.
Nome JNDI di destinazione
Il nome JNDI specificato quando si crea un argomento nel passo 5d.
Fare clic su **Avanti**.
 - h) Nel passo 6 della procedura guidata, **Associa riferimenti risorsa alle risorse**, immettere i dettagli nel campo **Nome JNDI della risorsa di destinazione** . Questo nome è il nome JNDI specificato per il factory di connessione della coda di elementi respinti nel passaggio 7c. Fare clic su **Avanti**.
 - i) Nel passo 7 della procedura guidata, **Associa i riferimenti delle voci dell'ambiente delle risorse alle risorse**, immettere i dettagli nel campo **Nome JNDI risorsa di destinazione** . Questo è il nome JNDI della coda di elementi respinti creata nel passo 8d. Fare clic su **Avanti**.
 - j) Nel passo 8 della procedura guidata, **Associa voci di ambiente per i moduli EJB**, accettare il valore predefinito 1. Fare clic su **Avanti**.
- Oracle RAC:** quando ci si connette a Oracle Real Application Cluster, è necessario impostare il valore della proprietà `_numberOfFailedAttemptsBeforeReject` su **almeno 2**. Questa proprietà determina il numero di volte in cui il programma di registrazione tenta di elaborare un messaggio di controllo dopo un errore. In caso di failover del database, è probabile che si verifichi almeno un malfunzionamento. Per evitare di spostare inutilmente un messaggio nella coda di elementi respinti, l'aumento di questo valore consente di effettuare un secondo tentativo, che di solito ha esito positivo quando viene effettuata una connessione alla nuova istanza del database. Se durante il test si rileva che i messaggi vengono ancora spostati nella coda di elementi respinti

durante il failover dell'istanza del database, aumentare ulteriormente questo valore: la tempistica del passaggio tra le istanze potrebbe causare più di un errore per lo stesso messaggio. Tuttavia, tenere presente che l'aumento di questo valore influisce su tutti i casi di errore (ad esempio, un messaggio in formato non corretto) e non solo sul failover del database, quindi aumentare il valore con attenzione per evitare tentativi non necessari.

k) Nel passo 9 della procedura guidata, **Metadata for modules**, fare clic su **Next**.

l) Nel passo 10 della procedura guidata, **Riepilogo**, fare clic su **Fine**.

10. È ora possibile avviare l'applicazione dalla console di gestione WebSphere Application Server traditional 9.0 :

a) Selezionare **Applicazioni > Tipi di applicazioni > WebSphere WebSphere** dalla navigazione della console.

b) Selezionare la casella di controllo per l'applicazione enterprise **Logger** dalla tabella di raccolta e fare clic su **Avvia**.

Configurazione dell'accesso utente per il logger del database Java EE per MFT

Quando si configura il programma di registrazione database Java Platform, Enterprise Edition (Java EE) per Managed File Transfer, sono necessari gli account utente per accedere a IBM MQ, al database e al sistema operativo. Il numero di utenti del sistema operativo richiesto dipende dal numero di sistemi utilizzati per ospitare questi componenti.

Informazioni su questa attività

Il numero e il tipo di account utente necessari per eseguire il programma di registrazione database Java EE dipende dal numero di sistemi utilizzati. Gli account utente sono necessari per accedere ai seguenti tre ambienti:

- Sistema operativo locale
- IBM MQ
- Database

È possibile installare il programma di registrazione database JEE, IBM MQ e il proprio database su un singolo sistema o su più sistemi. I componenti possono essere installati nelle seguenti topologie di esempio:

Java EE database logger, IBM MQ, e il database tutti sullo stesso sistema

È possibile definire un singolo utente del sistema operativo da utilizzare con tutti e tre i componenti. Il programma di registrazione utilizza la modalità Bindings per connettersi a IBM MQ e una connessione nativa per connettersi al database.

Java EE programma di registrazione database e IBM MQ su un sistema, il database su un sistema separato

Si creano due utenti per questa configurazione: un utente del sistema operativo sul sistema che esegue il programma di registrazione e un utente del sistema operativo con accesso remoto al database sul server database. Il programma di registrazione utilizza la modalità Bindings per connettersi a IBM MQ e una connessione client per accedere al database.

Java EE programma di registrazione database su un sistema, IBM MQ su un altro sistema, il database su un altro sistema

Si creano tre utenti per questa configurazione: un utente del sistema operativo per avviare il server delle applicazioni, un utente IBM MQ per accedere alle code e agli argomenti utilizzati e un utente del server di database per accedere e inserire nelle tabelle del database. Il programma di registrazione utilizza la modalità Client per accedere a IBM MQ e una connessione client per accedere al database.

Come esempio, il resto di queste istruzioni presuppone che l'utente sia denominato `fteLog`, ma è possibile utilizzare qualsiasi nome utente, nuovo o esistente. Configurare le autorizzazioni utente come segue:

Procedura

1. Verificare che l'utente del sistema operativo disponga di un proprio gruppo e che non appartenga ad alcun gruppo con ampie autorizzazioni sul gestore code di coordinamento. L'utente non deve essere nel gruppo mqm. Su alcune piattaforme, al gruppo di staff viene automaticamente fornito l'accesso al gestore code; l'utente del programma di registrazione non deve far parte del gruppo di staff. È possibile visualizzare i record di autorizzazioni per il gestore code stesso e per gli oggetti in esso contenuti utilizzando IBM MQ Explorer. Fare clic con il pulsante destro del mouse sull'oggetto e selezionare **Autorizzazioni oggetto > Gestisci record di autorizzazione**. Nella riga di comando, è possibile utilizzare i comandi `dspmqaout` (visualizza autorità) o `dmpmqaut` (dump autorità).
2. Utilizzare la finestra **Gestisci record di autorizzazioni** in IBM MQ Explorer o il comando `setmqaut` (concessione o revoca dell'autorità) per aggiungere autorizzazioni per il proprio gruppo di utenti IBM MQ (in AIX, IBM MQ le autorizzazioni sono associate solo a gruppi, non a singoli utenti). Le autorità richieste sono le seguenti:
 - CONNECT e INQUIRE sul gestore code (le librerie IBM MQ Java richiedono l'autorizzazione INQUIRE per funzionare).
 - Autorizzazione SUBSCRIBE su SYSTEM.FTE .
 - Autorizzazione PUT su SYSTEM.FTE.LOG.RJCT.*nome_programma di registrazione* .I nomi delle code di comando e di rifiuto forniti sono i nomi predefiniti. Se sono stati scelti nomi di coda differenti quando sono state configurate le code del programma di registrazione, aggiungere le autorizzazioni a tali nomi di coda.
3. Eseguire la configurazione dell'utente del database specifica per il database che si sta utilizzando.
 - Se il database è Db2, effettuare le seguenti operazioni:

Nota: Esistono diversi meccanismi per la gestione degli utenti del database con Db2. Queste istruzioni si applicano allo schema predefinito in base agli utenti del sistema operativo.

 - Assicurarsi che l'utente `fte1log` non si trovi in alcun gruppo di gestione Db2 (ad esempio, `db2iadm1`, `db2fadm1` o `dasadm1`).
 - Fornire all'utente l'autorizzazione per la connessione al database e l'autorizzazione per selezionare, inserire e aggiornare le tabelle create come parte del [Passo 2: creare le tabelle di database richieste](#).
 - Se il database è Oracle, effettuare le seguenti operazioni:
 - Assicurarsi che l'utente `fte1log` non sia in alcun gruppo di amministrazione Oracle (ad esempio, `ora_dba` su Windows o `dba` su AIX and Linux).
 - Fornire all'utente l'autorizzazione a collegarsi al database e l'autorizzazione a selezionare, inserire e aggiornare le tabelle create come parte del [Passo 2: creare le tabelle di database richieste](#).

Migrazione dal programma di registrazione database autonomo al programma di registrazione database di Java EE per MFT

È possibile eseguire la migrazione dal programma di registrazione database autonomo al programma di registrazione database Java EE . È necessario arrestare il programma di registrazione database autonomo e installare il programma di registrazione database JEE. Per evitare la perdita o la duplicazione delle voci di log, è necessario arrestare la pubblicazione dei messaggi nel sistema SYSTEM.FTE prima di arrestare il programma di registrazione database autonomo e riavviarlo dopo aver installato il programma di registrazione database Java EE . Eseguire il backup del database prima della migrazione.

Informazioni su questa attività

Procedura


1. Prima di interrompere il database, eseguire questo comando MQSC rispetto al proprio gestore code di coordinamento: `ALTER QM PSMODE (COMPAT)`

Questo arresta la pubblicazione dei messaggi nel SISTEMA SYSTEM.FTE/Log . Attendere che il programma di registrazione abbia elaborato tutti i messaggi sulla relativa sottoscrizione. Per impostazione predefinita, questa sottoscrizione è denominata SYSTEM.FTE.LOGGER.AUTOAUTO.

2. Arrestare il programma di registrazione database utilizzando il comando **fteStopLogger** .
3. Eseguire il backup del database utilizzando gli strumenti forniti con il software del database.
4. Eliminare la sottoscrizione appartenente al programma di registrazione database autonomo.

Per impostazione predefinita, questa sottoscrizione è denominata SYSTEM.FTE.LOGGER.AUTOAUTO.

5. Se lo schema del database è una versione precedente, è necessario migrare lo schema a ciascun livello successivo in ordine. Ad esempio, se lo schema del database è V7.0.1 e si sta eseguendo la migrazione a V7.0.4, è necessario migrare lo schema da V7.0.1 a V7.0.2, quindi da V7.0.2 a V7.0.3e quindi da V7.0.3 a V7.0.4. Migrare lo schema del database dalla versione *old* alla versione *new*, dove *old* e *new* sono le variabili che descrivono una versione dello schema, eseguendo una delle seguenti azioni per ciascuna versione dello schema da migrare:

-  Se il database è Db2 su z/OS e si sta eseguendo la migrazione tra schemi V7.0.2 e V7.0.3 o tra gli schemi V7.0.3 e V7.0.4 , è necessario creare un nuovo schema di database e copiare i dati esistenti in esso. Per ulteriori informazioni, consultare la documentazione Db2.
- Se il tuo database non è Db2 o hai creato il tuo database con una dimensione di pagina superiore a 8K, puoi migrare lo schema nello stesso modo come per le altre versioni, completando la seguente procedura.
- Se si sta eseguendo la migrazione tra le tabelle del database in altre circostanze, completare la seguente procedura:
 - a. Scegliere il file appropriato per la piattaforma del database e che abbia un nome che includa la stringa *old-new*. Questo file si trova nella directory *MQ_INSTALLATION_PATH/mqft/sql* dell'installazione di Strumenti remoti e documentazione.
 - b. Se sono state apportate modifiche allo schema iniziale, esaminare il file di migrazione per assicurarsi che sia compatibile con il database modificato.
 - c. Eseguire il file SQL sul proprio database.

6. Installare il file EAR del programma di registrazione database Java EE .
7. Distribuire il programma di registrazione del database Java EE . Per ulteriori informazioni, consultare [“Installazione del programma di registrazione database Java EE per MFT”](#) a pagina 822.
8. Eseguire il seguente comando MQSC rispetto al gestore code di coordinamento: ALTER QMGR PSMODE(ENABLED)

Ciò consente la pubblicazione dei messaggi nel SISTEMA SYSTEM.FTE/Log .

Risultati

Configurazione del bridge Connect:Direct

Configurare il bridge Connect:Direct per trasferire i file tra una rete Managed File Transfer e una rete Connect:Direct . I componenti del bridge Connect:Direct sono un nodo Connect:Direct e un agent Managed File Transfer dedicati alla comunicazione con tale nodo. A questo agent si fa riferimento come agent bridge Connect:Direct .

Prima di iniziare

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.

Per ulteriori dettagli sulle versioni del sistema operativo supportate per il bridge Connect:Direct , consultare la pagina Web [Requisiti di sistema per IBM MQ](#).

Informazioni su questa attività

Un agent bridge Connect:Direct è un agent Managed File Transfer dedicato alla comunicazione con un nodo Connect:Direct .

Per impostazione predefinita, l'agent bridge Connect:Direct utilizza il protocollo TCP / IP per connettersi al nodo Connect:Direct . Se si desidera una connessione sicura tra l'agent bridge Connect:Direct e il nodo Connect:Direct , è possibile utilizzare il protocollo SSL o il protocollo TLS.

Procedura

1. Scegliere i sistemi operativi per il nodo e l'agent bridge Connect:Direct :

- a) Scegliere un sistema su cui è in esecuzione Windows o Linux su x86-64 su cui installare l'agent bridge Connect:Direct .
- b) Scegliere un sistema operativo supportato da Connect:Direct per Windows o Connect:Direct per UNIX su cui installare il nodo bridge Connect:Direct .

2. Scegliere e configurare un nodo Connect:Direct .

È necessario avere un nodo Connect:Direct installato prima di seguire queste istruzioni.

- a) Scegliere un nodo Connect:Direct con cui comunicare con l'agent Managed File Transfer .
- b) Controllare la mappa di rete per il nodo Connect:Direct scelto. Se la mappa di rete contiene voci per i nodi remoti in esecuzione su un sistema operativo Windows , è necessario assicurarsi che tali voci specifichino che i nodi sono in esecuzione su Windows.



Se il nodo Connect:Direct selezionato per il bridge Connect:Direct è in esecuzione su Windows, utilizzare Connect:Direct Requester per modificare la mappa di rete. Assicurarsi che il campo **Sistema operativo** per tutti i nodi remoti in esecuzione su Windows sia impostato su **Windows**.

3. Creare e configurare un agent bridge Connect:Direct .

- a) Creare un agent bridge Connect:Direct utilizzando il comando **fteCreateCDAgent** .
 - È necessario fornire un valore per il parametro **cdNode** . Questo parametro specifica il nome utilizzato dall'agent per il nodo Connect:Direct che fa parte del bridge Connect:Direct . Utilizzare il nome del nodo Connect:Direct scelto nella sezione precedente.
 - Fornire i parametri **cdNodeHost** e **cdNodePort** , che definiscono il nodo Connect:Direct con cui comunica l'agent.

Se non si fornisce un valore per il parametro **cdNodeHost** , viene utilizzato il nome host o l'indirizzo IP del sistema locale. Se non si fornisce un valore per il parametro **cdNodePort** , viene utilizzato il valore 1363 .


- Facoltativamente, utilizzare le informazioni in [fteCreateAgent](#) per stabilire se è necessario specificare un valore per il parametro **cdTmpDir**.
 - b) Associare le credenziali utente utilizzate da Managed File Transfer alle credenziali utente su un nodo Connect:Direct. È possibile associare le credenziali utilizzando uno dei seguenti metodi:
 - Creare un file `ConnectDirectCredentials.xml` per definire le informazioni di associazione credenziali. Per ulteriori informazioni, consultare [“Associazione delle credenziali per Connect:Direct utilizzando il file ConnectDirectCredentials.xml”](#) a pagina 835.
 - Scrivere un'uscita utente per eseguire l'associazione credenziali per il bridge Connect:Direct. Per ulteriori informazioni, consultare [“Associazione delle credenziali per Connect:Direct utilizzando le classi di uscita”](#) a pagina 837.
4. Configurare il file `ConnectDirectNodeProperties.xml` per includere informazioni sui nodi Connect:Direct remoti.

È necessario aver creato un agent bridge Connect:Direct prima di seguire queste istruzioni.

Modificare il modello `ConnectDirectNodeProperties.xml` nella directory di configurazione dell'agente bridge Connect:Direct. Per ogni nodo o gruppo di nodi Connect:Direct di cui si desidera definire le informazioni, effettuare le seguenti operazioni:

- a) All'interno dell'elemento `nodeProperties`, creare un elemento `node`.
- b) Aggiungere un attributo `name` all'elemento `node`. Specificare il valore di questo attributo come un modello che corrisponda al nome di uno o più nodi Connect:Direct remoti.
- c) Opzionale: Aggiungere un attributo `pattern` all'elemento `node` che specifica quale tipo di pattern è il valore nell'attributo `name`. I valori validi sono `regex` e `wildcard`. L'opzione predefinita è `wildcard`.
- d) Aggiungere un attributo `type` all'elemento `node` che specifica il sistema operativo su cui vengono eseguiti i nodi Connect:Direct remoti specificati dall'attributo `name`.

I valori validi sono:

- Windows - il nodo viene eseguito su Windows
- UNIX - il nodo viene eseguito su AIX and Linux
-  z/OS, zos, os/390o os390 - il nodo viene eseguito su z/OS

Il valore di questo attributo non è sensibile al maiuscolo / minuscolo. I trasferimenti ai nodi remoti che si trovano su altri sistemi operativi non sono supportati dal bridge Connect:Direct.

Per ulteriori informazioni, vedere [Connect:Direct formato file delle proprietà del nodo](#).

5. Configurare una connessione protetta tra l'agent bridge Connect:Direct e il nodo Connect:Direct. Per un esempio di come eseguire questa operazione, consultare [Configurazione di SSL o TLS tra l'agent bridge Connect:Direct e il nodo Connect:Direct](#).


Attività correlate

[Risoluzione dei problemi relativi al bridge Connect:Direct](#)

[Configurazione di SSL o TLS tra l'agent bridge Connect:Direct e il nodo Connect:Direct](#)

[Trasferimento di un file a un nodo Connect:Direct](#)

[Trasferimento di un file da un nodo Connect:Direct](#)

 [Trasferimento di più file da un nodo Connect:Direct](#)

Riferimenti correlati

[Il bridge Connect:Direct](#)

Associazione delle credenziali per Connect:Direct

Associare le credenziali utente in Managed File Transfer con le credenziali utente su un nodo Connect:Direct utilizzando la funzione di associazione credenziali predefinita dell'agent bridge Connect:Direct o scrivendo la propria uscita utente. Managed File Transfer fornisce un'uscita utente di esempio che esegue l'associazione delle credenziali utente.

Attività correlate

[“Associazione delle credenziali per Connect:Direct utilizzando il file ConnectDirectCredentials.xml” a pagina 835](#)

Associare le credenziali utente in Managed File Transfer alle credenziali utente sui nodi Connect:Direct utilizzando la funzione di associazione credenziali predefinita dell'agent bridge Connect:Direct . Managed File Transfer fornisce un file XML che è possibile modificare per includere le informazioni sulle credenziali.

[“Associazione delle credenziali per Connect:Direct utilizzando le classi di uscita” a pagina 837](#)

Se non si desidera utilizzare la funzione di associazione credenziali predefinita dell'agent bridge Connect:Direct , è possibile associare le credenziali utente in Managed File Transfer alle credenziali utente su un nodo Connect:Direct scrivendo la propria uscita utente. La configurazione delle proprie uscite utente di associazione credenziali disabilita la funzione di associazione credenziali predefinita.

Riferimenti correlati

[Interfaccia CDCredentialExit.java](#)

[Formato file delle credenziali Connect:Direct](#)

Associazione delle credenziali per Connect:Direct utilizzando il file ConnectDirectCredentials.xml

Associare le credenziali utente in Managed File Transfer alle credenziali utente sui nodi Connect:Direct utilizzando la funzione di associazione credenziali predefinita dell'agent bridge Connect:Direct . Managed File Transfer fornisce un file XML che è possibile modificare per includere le informazioni sulle credenziali.

Informazioni su questa attività

Dopo che un agent bridge Connect:Direct è stato creato utilizzando il comando **fteCreateCDAgent** , è necessario creare manualmente un file `ConnectDirectCredentials.xml` . Prima di poter utilizzare un agent bridge Connect:Direct , è necessario modificare questo file per includere le informazioni su host, utente e credenziali. Per ulteriori informazioni, vedi [Formato del file delle credenziali Connect:Direct](#) . Per impostazione predefinita, questo file viene caricato dalla directory home dell'utente corrente, ad esempio `/home/fteuser/ConnectDirectCredentials.xml` . Per utilizzare un'altra posizione, specificarla utilizzando l'elemento `<credentialsFile>` nel file `ConnectDirectNodeProperties.xml` .

Procedura

1. Verificare che l'attributo `name` nell'elemento `<tns:pnode name="Connect:Direct node host" pattern="wildcard">` contenga il valore del nome del nodo Connect:Direct a cui si connette l'agent bridge Connect:Direct . Questo valore deve essere lo stesso valore specificato per il parametro **fteCreateCDAgent -cdNode** .

Il valore dell'attributo `pattern` può essere `wildcard` o `regex` . Se questo attributo non viene specificato, il valore predefinito è `wildcard` .

2. Inserire le informazioni sulle credenziali e sull'ID utente nel file come elementi child di `<tns:pnode>` . È possibile inserire una o più istanze del seguente elemento `<tns:user>` nel file:

```
<tns:user name="name"
          pattern="pattern"
          ignorecase="ignorecase"
          cdUserId="cdUserId"
          cdPassword="cdPassword"
          pnodeUserId="pnodeUserId"
          pnodePassword="pnodePassword">
</tns:user>
```

dove:

- `name` è un modello che corrisponde all'ID utente MQMD associato alla richiesta di trasferimento MFT .

- *pattern* specifica se il modello specificato per l'attributo name è un'espressione jolly o un'espressione regolare Java . Il valore dell'attributo pattern può essere wildcard o regex. Se questo attributo non viene specificato, il valore predefinito è wildcard.
- *ignorecase* specifica se considerare il modello specificato dall'attributo name come sensibile al maiuscolo / minuscolo. Se questo attributo non viene specificato, il valore predefinito è true.
- *cdUserId* è l'ID utente utilizzato dall'agent bridge Connect:Direct per connettersi al nodo Connect:Direct specificato dall'elemento name di <tns : pnode> . Se possibile, assicurarsi che *cdUserId* sia un ID utente amministratore Connect:Direct . Se *cdUserId* non può essere un amministratore Connect:Direct , assicurarsi che l'ID utente disponga delle seguenti autorizzazioni funzionali sul nodo bridge Connect:Direct :
 - Per un nodo Windows impostare le seguenti autorizzazioni. Questo esempio è formattato con ritorni a capo per facilitare la leggibilità:

```
View Processes in the TCQ      value: yes
Issue the copy receive, copy send, run job, and run task
Process statements          value: yes
Issue the submit Process statement
Monitor, submit, change, and delete all Processes
Access Process statistics    value: all
Use the trace tool or issue traceon and traceoff commands
Override Process options such as file attributes and remote node ID      value: yes
```

- Per un nodo AIX o Linux impostare i seguenti parametri nel file `userfile.cfg` :

```
pstmt.copy                  value: y
pstmt.upload                value: y
pstmt.download              value: y
pstmt.runjob                value: y
pstmt.runtask               value: y
cmd.submit                  value: y
pstmt.submit                value: y
cmd.chgproc                 value: y
cmd.delproc                 value: y
cmd.flsproc                 value: y
cmd.selproc                 value: a
cmd.selstats                value: a
cmd.trace                   value: y
snode.ovrd                  value: y
```

- *cdPassword* è la password associata all'ID utente specificato dall'attributo `cdUserId` .
- Facoltativamente, è possibile specificare l'attributo `pnodeUserId` . Il valore di questo attributo è l'ID utente utilizzato dal nodo Connect:Direct specificato dall'attributo name dell'elemento <tns : pnode> per inoltrare il processo Connect:Direct . Se non si specifica l'attributo `pnodeUserId` , il nodo Connect:Direct utilizza l'ID utente specificato dall'attributo `cdUserId` per inoltrare il processo Connect:Direct .
- Facoltativamente, è possibile specificare l'attributo `pnodePassword`. Il valore di questo attributo è la parola d'ordine associata all'id utente specificato dall'attributo `pnodeUserId` .

Se nessun elemento utente corrisponde all'ID utente MQMD, il trasferimento ha esito negativo.

3. Opzionale: È possibile includere uno o più elementi <tns : snode> come elementi child dell'elemento <tns : user> . L'elemento <tns : snode> specifica le credenziali utilizzate dal nodo Connect:Direct che fa parte del bridge Connect:Direct . Queste credenziali sono l'ID utente e la password che il nodo

bridge Connect:Direct utilizza per connettersi al nodo Connect:Direct che è l'origine o la destinazione del trasferimento file.

Inserire uno o più dei seguenti elementi nel file:

```
<tns:snode name="name"
           pattern="pattern"
           userId="userId"
           password="password" />
```

dove:

- *name* è un modello che corrisponde al nome del nodo Connect:Direct che è l'origine o la destinazione del trasferimento file.
- *pattern* specifica se il modello specificato per l'attributo name è un'espressione jolly o un'espressione regolare Java . Il valore dell'attributo pattern può essere wildcard o regex. Se questo attributo non viene specificato, il valore predefinito è wildcard.
- *userId* è l'ID utente utilizzato dal nodo Connect:Direct specificato dall'attributo name dell'elemento <tns:snode> per connettersi a un nodo Connect:Direct che corrisponde al modello specificato dall'attributo name di <tns:snode>.
- *password* è la password associata all'ID utente specificato dall'attributo userId .

Se nessun elemento <tns:snode> corrisponde al nodo secondario del trasferimento file, il trasferimento non avrà esito negativo. Il trasferimento è stato avviato e non è stato specificato alcun ID utente e password da utilizzare con snode.

Risultati

Durante la ricerca di una corrispondenza di modello per i nomi utente o i nomi nodo Connect:Direct , 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” a pagina 832](#)

Configurare il bridge Connect:Direct per trasferire i file tra una rete Managed File Transfer e una rete Connect:Direct . I componenti del bridge Connect:Direct sono un nodo Connect:Direct e un agent Managed File Transfer dedicati alla comunicazione con tale nodo. A questo agent si fa riferimento come agent bridge Connect:Direct .

Riferimenti correlati

[Formato file delle credenziali Connect:Direct](#)

[fteCreateCDAgent: crea un agent bridge Connect:Direct](#)

Associazione delle credenziali per Connect:Direct utilizzando le classi di uscita

Se non si desidera utilizzare la funzione di associazione credenziali predefinita dell'agent bridge Connect:Direct , è possibile associare le credenziali utente in Managed File Transfer alle credenziali utente su un nodo Connect:Direct scrivendo la propria uscita utente. La configurazione delle proprie uscite utente di associazione credenziali disabilita la funzione di associazione credenziali predefinita.

Informazioni su questa attività

Le uscite utente create per associare le credenziali Connect:Direct devono implementare l'interfaccia `com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit`. Per ulteriori informazioni, vedi [CDCredentialExit.java](#) interface.

Configurazione di IBM MQ Console e REST API

Il server mqweb che ospita IBM MQ Console e REST API viene fornito con una configurazione predefinita. Per poter utilizzare uno di questi componenti è necessario completare una serie di attività

di configurazione, come la configurazione della sicurezza per consentire agli utenti di accedere. Questo argomento descrive tutte le opzioni di configurazione disponibili.

Procedura

- [“Configurazione di base per il server mqweb” a pagina 838](#)
- [“Configurazione della sicurezza” a pagina 843](#)
- [“Configurazione del nome host HTTP” a pagina 844](#)
- [“Configurazione delle porte HTTPS e HTTP” a pagina 845](#)
- [“Configurazione del timeout di risposta” a pagina 846](#)
- [“Configurazione dell'avvio automatico” a pagina 847](#)
- [“Configurazione della registrazione nei log” a pagina 848](#)
- [“Configurazione del token LTPA” a pagina 851](#)
- [“Configurazione del comportamento della connessione del gestore code remoto per IBM MQ Console” a pagina 853](#)
- [“Configurazione del gateway administrative REST API” a pagina 855](#)
- [“Configurazione di messaging REST API” a pagina 857](#)
- [“Configurazione di REST API per MFT” a pagina 863](#)
- [“Ottimizzazione della JVM del server mqweb” a pagina 868](#)
- [“Struttura del file del componente di installazione IBM MQ Console e REST API” a pagina 870](#)





Configurazione di base per il server mqweb

Prima di poter iniziare a utilizzare REST API o IBM MQ Console, è necessario installare i componenti corretti e configurare il server mqweb che ospita REST API o IBM MQ Console.

Informazioni su questa attività



La procedura per questa attività si concentra su una configurazione di base per il server mqweb in modo da poter iniziare rapidamente con REST API e IBM MQ Console. La procedura per la configurazione della sicurezza descrive come impostare un registro utente di base, ma esistono altre opzioni per la configurazione di utenti e ruoli. Per ulteriori informazioni sulla configurazione della sicurezza per il server mqweb, consultare [Sicurezza IBM MQ Console e REST API](#).

Nota: Per completare questa procedura, è necessario avere accesso al file `mqwebuser.xml` :

-  Su z/OS, è necessario essere un utente con accesso in scrittura al file `mqwebuser.xml` .
-  Su tutti i sistemi operativi, è necessario essere un utente privilegiato per accedere al file `mqwebuser.xml` .
-   Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server , è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server .

Procedura

1. Installare il componente IBM MQ Console e REST API :

-  Su AIX, installare il fileset `mqm.web.rte` . Per ulteriori informazioni sull'installazione dei sottopacchetti su AIX, vedere [Attività di installazione di AIX](#).
-  Su IBM i, installare il componente WEB. Per utilizzare questa funzione, è necessario installare anche i servizi di messaggistica e Web di 5724L26 IBM MQ Java e i prerequisiti di

5770JV1 Java SE 8. Per ulteriori informazioni sull'installazione delle funzioni su IBM i, vedere [Attività di installazione di IBM i](#).

- ▶ **Linux** Su Linux, installare il componente MQSeriesWeb . Per ulteriori informazioni sull'installazione dei componenti su Linux, consultare [Attività di installazione di Linux](#).
- ▶ **V 9.4.0** Da IBM MQ 9.4.0, è anche possibile eseguire il server mqweb in un'installazione IBM MQ Web Server autonoma su Linux. Per ulteriori informazioni, sull'installazione di IBM MQ Web Server, consultare [Installazione di IBM MQ Web Server autonomo](#).
- ▶ **Windows** Su Windows, installare la funzione Web Administration . Per ulteriori informazioni sull'installazione delle funzioni su Windows, vedere [Attività di installazione di Windows](#).
- ▶ **z/OS** Installare la funzione IBM MQ for z/OS UNIX System Services Web Components. Per ulteriori informazioni sull'installazione di componenti e funzioni su z/OS, consultare [Attività di installazione di z/OS](#).

2. Creare il server mqweb che ospita IBM MQ Console e REST API.

- ▶ **z/OS** Su z/OS, eseguire lo script **crtmqweb** .
Questo script crea la directory utente WebSphere Liberty contenente la configurazione del server mqweb e i file di log. Per ulteriori informazioni sull'esecuzione dello script **crtmqweb** , consultare [“Creazione del server mqweb”](#) a pagina 950.
- ▶ **V 9.4.0** ▶ **Linux** In un'installazione autonoma di IBM MQ Web Server , attenersi alla procedura descritta in [“Configurazione del IBM MQ Web Server autonomo”](#) a pagina 842.
- In tutti gli altri ambienti, non è necessario completare alcuna azione per creare il server mqweb.

3. ▶ **z/OS**

Su z/OS, creare una procedura catalogata per avviare il server mqweb.

Per ulteriori informazioni, consultare [“Creating a procedure for the mqweb server”](#) a pagina 952.

4. Sostituire il file di configurazione esistente, mqwebuser.xml , con il file di esempio del registro di base configurato per offrire la sicurezza di base. Copiare il file basic_registry.xml dalla directory `MQ_INSTALLATION_PATH/web/mq/samp/configuration` nella directory appropriata per il sistema e ridenominare il file in mqwebuser.xml:

- In un'installazione IBM MQ , copiare il file nella seguente directory:

– ▶ **Linux** ▶ **AIX** Su AIX and Linux: `/var/mqm/web/installations/installationName/servers/mqweb`

– ▶ **Windows** Su Windows:
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`

Dove `MQ_DATA_PATH` è il percorso dati IBM MQ , questo percorso è il percorso dati selezionato durante l'installazione di IBM MQ. Per impostazione predefinita, questo è `C:\ProgramData\IBM\MQ`.

– ▶ **z/OS** Su z/OS: `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.

- ▶ **V 9.4.0** ▶ **Linux** In un'installazione autonoma di IBM MQ Web Server :
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb`
dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente **MQ_OVERRIDE_DATA_PATH** .

Il file di esempio `basic_registry.xml` configura quattro utenti:

MQADMIN

Un utente amministrativo che è un membro del ruolo MQWebAdmin .

mqreader

Un utente di gestione di sola lettura che è membro del ruolo MQWebAdminRO .

mftadmin

Un utente amministrativo che è un membro del ruolo MFTWebAdmin .

mftreader

Un utente di gestione di sola lettura che è membro del ruolo MFTWebAdminRO .

Tutti gli utenti sono anche membri del ruolo MQWebUser .

Per ulteriori informazioni sui ruoli disponibili, consultare [Ruoli su IBM MQ Console e REST API](#)

5. Opzionale: Modificare il file `mqwebuser.xml` per aggiungere ulteriori utenti e gruppi. Assegnare a tali utenti e gruppi i ruoli appropriati per essere autorizzati ad utilizzare REST API o IBM MQ Console. È inoltre possibile modificare le password per gli utenti definiti per impostazione predefinita e codificare le nuove password. Per ulteriori informazioni, vedi [Configurazione di utenti e ruoli](#).

Nota:

- **z/OS** Su z/OS, se si aggiungono utenti al ruolo MQWebUser , è necessario concedere anche all'ID utente dell'attività avviata mqweb l'accesso alternativo agli ID utente con il ruolo MQWebUser . Ad esempio:

```
RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
```

- **Multi z/OS** Per completare la procedura per iniziare con messaging REST API, devi aggiungere un utente al file `mqwebuser.xml` . Questo utente deve avere lo stesso nome di un utente IBM MQ esistente sul sistema. Seguendo lo stesso formato degli altri utenti nel file XML, aggiungere l'ID utente e una password dopo la seguente riga nel file XML: `<user name="mftreader" password="mftreader"/>`.
6. Impostare l'ambiente in modo che punti alla configurazione del server mqweb.
- **z/OS** Su z/OS, impostare la variabile di ambiente WLP_USER_DIR in modo che la variabile 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 al comando `crtmqweb` . Ad esempio:

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

Per ulteriori informazioni, consultare [“Creazione del server mqweb” a pagina 950](#).

- **V9.4.0 Linux** In una installazione IBM MQ Web Server autonoma, impostare la variabile di ambiente **MQ_OVERRIDE_DATA_PATH** sulla directory di dati IBM MQ Web Server . Ad esempio, se si sceglie di utilizzare `/var/mqweb` come directory di dati IBM MQ Web Server , immettere il seguente comando:
- ```
export MQ_OVERRIDE_DATA_PATH=/var/mqweb
```
- In tutti gli altri ambienti, non è necessario completare alcuna azione per impostare l'ambiente.
7. Per default, REST API e IBM MQ Console sono disponibili solo dallo stesso host del server mqweb. Abilitare le connessioni remote al server mqweb immettendo il seguente comando:

```
setmqweb properties -k httpHost -v hostname
```



Dove *hostname* specifica l'indirizzo IP, il nome host DNS (domain name server) con suffisso nome dominio o il nome host DNS del server in cui è installato IBM MQ . Utilizzare un asterisco, \*, tra virgolette, per specificare tutte le interfacce di rete disponibili, come mostrato nel seguente esempio:

```
setmqweb properties -k httpHost -v "*"
```

8. Opzionale: Per impostazione predefinita, administrative REST API per MFT non è abilitato. Se si desidera utilizzare questa funzione, è necessario abilitarla e configurare un gestore code di coordinamento:

- a) Abilitare administrative REST API per MFT immettendo il seguente comando:

```
setmqweb properties -k mqRestMftEnabled -v true
```

- b) Configurare quale gestore code è il gestore code di coordinamento immettendo il seguente comando:

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

Dove *qmgrName* è il nome del gestore code di coordinamento.

- c) Per abilitare chiamate POST, configurare quale gestore code è il gestore code comandi immettendo il seguente comando:

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

Dove *qmgrName* è il nome del gestore code comandi.

9. Avviare il server mqweb che supporta REST API e IBM MQ Console:

- ▶ **ALW** Su AIX, Linux, and Windows, come utente privilegiato, immettere il seguente comando:

```
strmqweb
```

- ▶ **IBM i** Su IBM i, come utente privilegiato, immettere il seguente comando in Qshell:

```
/QIBM/ProdData/mqm/bin/strmqweb
```

- ▶ **z/OS** Su z/OS, avviare la procedura creata in [“Creating a procedure for the mqweb server”](#) a pagina 952.

I seguenti messaggi vengono emessi per STDOUT DD per indicare che il server mqweb è stato avviato correttamente.

```
[AUDIT] MQWB2019I: MQ Console level: 9.2.4 - V924-CD924-L211028
[AUDIT] MQWB0023I: MQ REST API level: 9.2.4 - V924-CD924-L211028
[AUDIT] CWWKZ0001I: Application com.ibm.mq.rest started in 1.763 seconds.
[AUDIT] CWWKZ0001I: Application com.ibm.mq.console started in 2.615 seconds.
[AUDIT] CWWKF0011I: The mqweb server is ready to run a smarter planet. The mqweb
server started in 10.016 seconds.
```

È possibile arrestare il server mqweb in qualsiasi momento arrestando l'attività avviata del server mqweb su z/OSo utilizzando il comando **endmqweb** . Tuttavia, se il server mqweb non è in esecuzione, non è possibile utilizzare REST API o IBM MQ Console.

10. ▶ **z/OS**

Opzionale: Su z/OS, se si desidera consentire ai prodotti di automazione del sistema di eseguire il trap dei messaggi MQWB2019I e MQWB0023I emessi all'avvio di IBM MQ Console e REST API , configurare il server mqweb per scrivere tali messaggi nella console MVS . Per configurare il server mqweb in modo che scriva i messaggi MQWB2019I e MQWB0023I nella console MVS , modificare il file mqwebuser.xml creato nel passo [“4”](#) a pagina 839e aggiungere la riga seguente al file:

```
<zosLogging enableLogToMVS="true" wtoMessage="MQWB2019I,MQWB0023I"/>
```

Per ulteriori informazioni sulla configurazione della registrazione z/OS nel server mqweb, consultare [z/OS Logging \(zosLogging\)](#).

## Operazioni successive

1. Configurare le impostazioni del server mqweb, inclusa l'abilitazione delle connessioni HTTP e la modifica del numero di porta. Per ulteriori informazioni, consultare [“Configurazione di IBM MQ Console e REST API”](#) a pagina 837.
2. Facoltativamente, configurare REST API:
  - a. Configurare la condivisione di risorse tra origini per REST API. Per impostazione predefinita, non puoi accedere a REST API dalle risorse web che non sono ospitate sullo stesso dominio di REST API. In altre parole, le richieste tra origini non sono abilitate. È possibile configurare CORS (Cross Origin Resource Sharing) per consentire richieste tra origini da URL specificati. Per ulteriori informazioni, consultare [Configurazione di CORS per REST API](#).
  - b. Configurare REST API per MFT. Per ulteriori informazioni, consultare [“Configurazione di REST API per MFT”](#) a pagina 863.
3. Utilizzare REST API o IBM MQ Console:
  - [Introduzione a administrative REST API](#)
  - [Introduzione a messaging REST API](#)
  - [Introduzione a IBM MQ Console](#)

V 9.4.0

Linux

## Configurazione del IBM MQ Web Server autonomo

Da IBM MQ 9.4.0, è possibile eseguire il server mqweb che ospita IBM MQ Console e REST API in una installazione IBM MQ Web Server autonoma.

### Prima di iniziare

Il IBM MQ Web Server autonomo è disponibile solo su Linux.

Prima di poter configurare il server mqWeb, è necessario installare IBM MQ Web Server attenendosi alla procedura in [Installazione del IBM MQ Web Server autonomo](#).

### Informazioni su questa attività

Seguire la procedura riportata in questa attività per creare e configurare un nuovo server mqweb che viene eseguito in un'installazione IBM MQ Web Server autonoma. È possibile configurare più di un server mqweb da eseguire in un'installazione IBM MQ Web Server autonoma ripetendo questa procedura.

### Procedura

1. Creare la directory di dati IBM MQ Web Server .

La directory di dati viene utilizzata per archiviare i file di configurazione e di log per il server mqweb che esegue IBM MQ Console e REST API. È possibile utilizzare qualsiasi directory scelta come directory di dati IBM MQ Web Server .

All'ID utente utilizzato per avviare il server mqweb deve essere concesso l'accesso in lettura e scrittura alla directory dei dati.

2. Impostare la variabile di ambiente di **MQ\_OVERRIDE\_DATA\_PATH** sulla directory di dati creata nel passo “1” a pagina 842.

Ad esempio, se si sceglie di utilizzare /var/mqweb come directory di dati IBM MQ Web Server , immettere il seguente comando:

```
export MQ_OVERRIDE_DATA_PATH=/var/mqweb
```

3. Utilizzare il comando **setmqenv** per impostare l'ambiente IBM MQ .

Passare alla directory bin della directory di installazione IBM MQ Web Server , quindi immettere il seguente comando:

```
. setmqenv -s
```

4. Utilizzare il comando **crtmqdir** per creare le directory e i file IBM MQ nella directory dei dati. I file creati includono una definizione di template per il server mqweb.

Emetti il seguente comando:

```
crtmqdir -s -f
```

5. Opzionale: Se questo server mqweb è il primo creato per essere eseguito con questa installazione del IBM MQ Web Server autonomo, utilizzare il comando **mqlicense** per esaminare e accettare la licenza IBM MQ .

È necessario eseguire questo comando come utente con accesso in scrittura alla directory di installazione IBM MQ Web Server .

Ad esempio, immettere il seguente comando per visualizzare la licenza IBM MQ :

```
mqlicense
```

Per ulteriori informazioni, consultare [mqlicense](#).

6. Opzionale: Per migrare un server mqweb esistente da eseguire nell'installazione autonoma di IBM MQ Web Server appena configurata, completare la seguente procedura:
  - a. Eseguire il backup della configurazione del server mqweb esistente.
  - b. Ripristinare i file nella directory `MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST` , dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server creata nel passo “1” a [pagina 842](#).

Per ulteriori informazioni, consultare [“Backup e ripristino della configurazione del server mqweb” a pagina 872](#).

**Nota:** Alcune funzioni di IBM MQ Console e REST API non sono disponibili in un'installazione IBM MQ Web Server autonoma. Se si migra un server mqweb da un'installazione IBM MQ a un'installazione IBM MQ Web Server autonoma, queste funzioni non possono essere utilizzate dopo la migrazione. Per ulteriori informazioni sulle limitazioni applicabili in un'installazione IBM MQ Web Server autonoma, consultare [IBM MQ Console e REST API](#).

## Operazioni successive

Configurare il server mqweb attenendosi alla procedura descritta in [“Configurazione di base per il server mqweb” a pagina 838](#).

## Configurazione della sicurezza

È possibile configurare la protezione per IBM MQ Console e REST API modificando il file `mqwebuser.xml` . È possibile configurare e autenticare gli utenti configurando un registro utenti di base o un registro LDAP o uno qualsiasi degli altri tipi di registro forniti con WebSphere Liberty. È quindi possibile autorizzare tali utenti assegnando un ruolo a utenti e gruppi.

## Informazioni su questa attività

Per configurare la sicurezza per IBM MQ Console e REST API, è necessario configurare utenti e gruppi. Questi utenti e gruppi possono quindi essere autorizzati ad utilizzare IBM MQ Console, REST API o entrambi. Per ulteriori informazioni sulla configurazione di utenti e gruppi e sull'autenticazione e l'autorizzazione degli utenti, consultare [Sicurezza IBM MQ Console e REST API](#).

Quando gli utenti eseguono l'autenticazione con IBM MQ Console, viene generato un token LTPA. Questo token consente all'utente di utilizzare IBM MQ Console senza eseguire nuovamente l'autenticazione fino alla scadenza del token.

Se si utilizza l'autenticazione basata su token con REST API, viene generato un token LTPA diverso quando l'utente accede utilizzando la risorsa `/login` REST API con il metodo HTTP POST. È possibile configurare quando questo token scade e se può essere utilizzato per entrambe le connessioni HTTP e HTTPS. Per ulteriori informazioni, consultare [“Configurazione del token LTPA” a pagina 851](#).

## Procedura




- [IBM MQ Console e sicurezza REST API](#)
- [“Configurazione del token LTPA” a pagina 851](#)

## Configurazione del nome host HTTP

Per impostazione predefinita, il server mqweb che ospita IBM MQ Console e REST API è configurato per consentire solo connessioni locali. Ovvero, è possibile accedere a IBM MQ Console e REST API solo sul sistema su cui sono installati IBM MQ Console e REST API. È possibile configurare il nome host per consentire le connessioni remote utilizzando il comando `setmqweb`.

### Prima di iniziare

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi `dspmqweb` e `setmqweb`:

-  Su z/OS, è necessario disporre dell'autorità per eseguire i comandi `dspmqweb` e `setmqweb` e l'accesso in scrittura al file `mqwebuser.xml`.
-  Su tutti gli altri sistemi operativi, è necessario essere un [utente privilegiato](#).
-  Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.



#### Attenzione:

Prima di emettere i comandi `setmqweb` o `dspmqweb` su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).



#### Attenzione:

Prima di immettere i comandi `setmqweb` o `dspmqweb` in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente `MQ_OVERRIDE_DATA_PATH` sulla directory di dati IBM MQ Web Server.

## Procedura

- Visualizzare la configurazione corrente del nome host HTTP utilizzando il seguente comando:  
`dspmqweb properties -a`  
Il campo `httpHost` mostra il nome host HTTP.
- Impostare il nome host HTTP utilizzando il seguente comando:

```
setmqweb properties -k httpHost -v hostName
```

dove *hostName* specifica l'indirizzo IP, il nome host DNS (domain name server) con suffisso nome dominio o il nome host DNS del server in cui è installato IBM MQ . Utilizzare un asterisco tra virgolette per specificare tutte le interfacce di rete disponibili. Utilizzare il valore `localhost` per consentire solo connessioni locali.

- Annullare l'impostazione del nome host HTTP utilizzando il seguente comando:

```
setmqweb properties -k httpHost -d
```




## Configurazione delle porte HTTPS e HTTP

Per default, il server mqweb che ospita IBM MQ Console e REST API utilizza la porta HTTPS 9443. La porta associata alle connessioni HTTP non è abilitata. È possibile abilitare la porta HTTP, configurare una porta HTTPS diversa o disabilitare la porta HTTP o HTTPS. È possibile configurare le porte utilizzando il comando **setmqweb** .

### Prima di iniziare

Se si abilita la porta HTTP e si utilizza l'autenticazione basata su token, è necessario abilitare lo stesso token LTPA da utilizzare per le connessioni HTTP e HTTPS. Per ulteriori informazioni, consultare [“Configurazione del token LTPA” a pagina 851](#).

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmqweb** e **setmqweb**:

-  Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmqweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
-  Su tutti gli altri sistemi operativi, è necessario essere un [utente privilegiato](#).
-  Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server , è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server .



#### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).



#### Attenzione:

Prima di immettere i comandi **setmqweb** o **dspmqweb** in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente `MQ_OVERRIDE_DATA_PATH` sulla directory di dati IBM MQ Web Server .



**Attenzione:** Per impostazione predefinita, il server mqweb richiede che i token LTPA siano protetti per tutte le richieste. Se il server mqweb è configurato per richiedere che i token LTPA siano protetti, non è possibile completare le seguenti azioni quando ci si connette alla porta HTTP:

- Accedi a IBM MQ Console.
- Utilizzare l'autenticazione basata su token con REST API.

Per consentire ai token LTPA di essere utilizzati dalle richieste HTTP, impostare il valore della proprietà **secureLTPA** su `false`. Per ulteriori informazioni, consultare [“Configurazione del token LTPA”](#) a pagina 851.

## Procedura

- Visualizzare la configurazione corrente delle porte HTTPS e HTTP utilizzando il seguente comando:  

```
dspmweb properties -a
```

Il campo `httpPort` mostra la porta HTTP e il campo `httpsPort` mostra la porta HTTPS.
- Abilitare o configurare la porta HTTP: utilizzando il seguente comando:
  - Abilitare o impostare la porta HTTP utilizzando il seguente comando:  

```
setmqweb properties -k httpPort -v portNumber
```

dove *portNumber* specifica la porta che si desidera utilizzare per connessioni HTTP. È possibile disabilitare la porta utilizzando il valore `-1`.
  - Reimpostare il valore porta HTTP sul valore predefinito di `-1` utilizzando il seguente comando:  

```
setmqweb properties -k httpPort -d
```
- Configurare la porta HTTPS:
  - Impostare il numero di porta HTTPS utilizzando il seguente comando:  

```
setmqweb properties -k httpsPort -v portNumber
```

dove *portNumber* specifica la porta che si desidera utilizzare per le connessioni HTTPS. È possibile disabilitare la porta utilizzando il valore `-1`.
  - Reimpostare il numero di porta HTTPS sul valore predefinito `9443` utilizzando il seguente comando:  




```
setmqweb properties -k httpsPort -d
```

## Configurazione del timeout di risposta

Per impostazione predefinita, IBM MQ Console e REST API sono in timeout se il tempo impiegato per inviare una risposta a un client è superiore a 30 secondi. È possibile configurare IBM MQ Console e REST API per utilizzare un valore di timeout differente utilizzando il comando **setmqweb**.

### Prima di iniziare

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmweb** e **setmqweb**:

-  Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
-  Su tutti gli altri sistemi operativi, è necessario essere un utente privilegiato.
-  Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.



#### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).



**Attenzione:** Linux V 9.4.0

Prima di immettere i comandi **setmqweb** o **dspmqweb** in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente **MQ\_OVERRIDE\_DATA\_PATH** sulla directory di dati IBM MQ Web Server .

## Procedura

- Visualizzare la configurazione corrente del timeout della richiesta utilizzando il seguente comando:  
`dspmqweb properties -a`  
Il campo `mqRestRequestTimeout` mostra il valore corrente per il timeout di risposta. Per ulteriori informazioni, consultare `dspmqweb properties`.
- Impostare il timeout della richiesta utilizzando il comando seguente:  
`setmqweb properties -k mqRestRequestTimeout -v timeout`  
dove *timeout* specifica il tempo, in secondi, prima del timeout.
- Reimpostare il timeout della richiesta sul valore predefinito di 30 secondi utilizzando il seguente comando:  
`setmqweb properties -k mqRestRequestTimeout -d`

## Configurazione dell'avvio automatico

Per impostazione predefinita, il IBM MQ Console viene avviato automaticamente all'avvio del server mqweb. È possibile configurare se IBM MQ Console e REST API vengono avviati automaticamente utilizzando il comando **setmqweb** .

### Prima di iniziare

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](#).
- **Linux V 9.4.0** Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server , è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server .



**Attenzione:**

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).



**Attenzione:** Linux V 9.4.0

Prima di immettere i comandi **setmqweb** o **dspmqweb** in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente **MQ\_OVERRIDE\_DATA\_PATH** sulla directory di dati IBM MQ Web Server .

## Procedura

- Visualizzare la configurazione corrente dell'avvio automatico utilizzando il seguente comando:

```
dspmqweb properties -a
```

Il campo `mqRestAutostart` mostra se REST API viene avviato automaticamente e il campo `mqConsoleAutostart` mostra se IBM MQ Console viene avviato automaticamente.

- Configurare se IBM MQ Console viene avviato automaticamente utilizzando il seguente comando:

```
setmqweb properties -k mqConsoleAutostart -v start
```

dove `start` è il valore `true` se si desidera che IBM MQ Console venga avviato automaticamente o `false` in caso contrario.

- Configurare se REST API viene avviato automaticamente utilizzando il seguente comando:

```
setmqweb properties -k mqRestAutostart -v start
```

dove `start` è il valore `true` se si desidera che REST API venga avviato automaticamente o `false` in caso contrario.

## Configurazione della registrazione nei log

È inoltre possibile configurare i livelli di registrazione, la dimensione massima del file di log e il numero massimo di file di log utilizzati dal server mqweb che ospita IBM MQ Console e REST API. Puoi configurare la registrazione utilizzando il comando **setmqweb** .

### Prima di iniziare

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.
- **Linux V 9.4.0** Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server , è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server .



**Attenzione:**

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).





**Attenzione:** Linux V 9.4.0

Prima di immettere i comandi **setmqweb** o **dspsmqweb** in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente **MQ\_OVERRIDE\_DATA\_PATH** sulla directory di dati IBM MQ Web Server .

## Informazioni su questa attività

Il server mqweb scrive messaggi di log e traccia nei seguenti file di log:

### console.log e messages.log

Questi file contengono messaggi emessi da IBM MQ Console, REST API e dal server mqweb che esegue questi componenti.

### trace.log

Questo file contiene la traccia per IBM MQ Console e REST API. La traccia viene scritta in questo file solo se la traccia è abilitata.

I file di log per il server mqweb si trovano in una delle directory seguenti:

- In un'installazione IBM MQ :

- Linux AIX Su AIX o Linux: `/var/mqm/web/installations/installationName/servers/mqweb/logs`

- Windows Su Windows:  
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb\logs`, dove `MQ_DATA_PATH` è il percorso dati IBM MQ . Questo percorso è il percorso dati selezionato durante l'installazione di IBM MQ. Per impostazione predefinita, questo è `C:\ProgramData\IBM\MQ`.

- z/OS Su z/OS: `WLP_user_directory/servers/mqweb/logs`

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

- V 9.4.0 Linux In un'installazione autonoma di IBM MQ Web Server :  
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb/logs`  
dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente **MQ\_OVERRIDE\_DATA\_PATH** .

I file di traccia di messaggistica per il codice REST API di messaggistica eseguito nel server mqweb si trovano in una delle seguenti directory:

- In un'installazione IBM MQ :

- Linux AIX Su AIX o Linux: `/var/mqm/web/installations/installationName/servers/mqweb`

- Windows Su Windows:  
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`, dove `MQ_DATA_PATH` è il percorso dati IBM MQ . Questo percorso è il percorso dati selezionato durante l'installazione di IBM MQ. Per impostazione predefinita, questo è `C:\ProgramData\IBM\MQ`.

- z/OS Su z/OS: `WLP_user_directory/servers/mqweb`

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

- V 9.4.0 Linux In un'installazione autonoma di IBM MQ Web Server :  
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb`  
dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente **MQ\_OVERRIDE\_DATA\_PATH** .

Per ulteriori informazioni sull'abilitazione della traccia per:

- REST API, consultare [Traccia di REST API](#)
- IBM MQ Console, consultare [Traccia di IBM MQ Console](#)

## Procedura

- Visualizzare la configurazione corrente della registrazione REST API utilizzando il seguente comando:  
`dspmqweb properties -a`
  - Il campo `maxTraceFileSize` mostra la dimensione massima del file di log
  - Il campo `maxTraceFiles` mostra il numero massimo di file di log
  - Il campo `traceSpec` mostra il livello di traccia utilizzato
  - Il campo `maxMsgTraceFileSize` visualizza la dimensione massima del file di traccia di messaggistica
  - Il campo `maxMsgTraceFiles` mostra il numero massimo di file di traccia di messaggistica
- Configurare la dimensione massima dei file `messages.log` e `trace.log`:
  - Impostare la dimensione massima del file di log utilizzando il comando seguente:  
`setmqweb properties -k maxTraceFileSize -v size`  
dove *size* specifica la dimensione, in MB, che ogni file di log può raggiungere.
  - Ripristinare la dimensione massima del file di log sul valore predefinito di 20 MB utilizzando il seguente comando:  
`setmqweb properties -k maxTraceFileSize -d`
- Configurare il numero massimo di file `messages.log` e `trace.log`:
  - Impostare il numero massimo di ciascun file di log utilizzando i seguenti comandi:  
`setmqweb properties -k maxTraceFiles -v max`  
dove *max* specifica il numero massimo di file.
  - Reimpostare il numero massimo di ciascun file di log sul valore predefinito 2 utilizzando il seguente comando:  
`setmqweb properties -k maxTraceFiles -d`
- Configurare la dimensione massima del file di traccia di messaggistica:
  - Impostare la dimensione massima del file di traccia della messaggistica utilizzando il seguente comando:  
`setmqweb properties -k maxMsgTraceFileSize -v size`  
dove *size* specifica la dimensione, in MB, che ogni file di traccia di messaggistica può raggiungere.
  - Reimpostare la dimensione massima del file di traccia di messaggistica sul valore predefinito di 200 MB utilizzando il seguente comando:  
`setmqweb properties -k maxMsgTraceFileSize -d`
- Configurare il numero massimo di file di traccia di messaggistica da utilizzare:
  - Impostare il numero massimo di file da utilizzare per la traccia di messaggistica utilizzando il seguente comando:  
`setmqweb properties -k maxMsgTraceFiles -v max`  
dove *max* specifica il numero massimo di file.
  - Reimpostare il numero massimo di file da utilizzare per la traccia di messaggistica sul valore predefinito 5 utilizzando il seguente comando:  
`setmqweb properties -k maxMsgTraceFiles -d`

- Configurare il livello di traccia che il server mqweb scrive:
  - Impostare la specifica di traccia utilizzata utilizzando il seguente comando:
 

```
setmqweb properties -k traceSpec -v level
```

 dove *livello* è uno dei valori elencati in Tabella 52 a pagina 851. La tabella delinea i livelli di registrazione, ordinati in base al livello di dettaglio crescente. Quando si abilita un livello di registrazione, si abilita anche ciascun livello prima di esso. Ad esempio, se si abilita il livello di registrazione **\*=warning**, si abilitano anche i livelli di registrazione **\*=severee** e **\*=fatal**.  
 Modificare questo valore quando richiesto dal supporto IBM .
  - Reimpostare la specifica di traccia utilizzata sul valore predefinito **\*=info** utilizzando il seguente comando:
 

```
setmqweb properties -k traceSpec -d
```

*Tabella 52. Livelli di registrazione validi*

Valore	Livello di registrazione applicato
* =disattivo	La registrazione è disattivata.
* =irreversibile	L'attività non può continuare e il componente, l'applicazione e il server non possono funzionare.
* =grave	L'attività non può continuare ma il componente, l'applicazione e il server possono ancora funzionare. Questo livello può anche indicare un errore irreversibile imminente.
* =avvertenza	Errore potenziale o imminente. Questo livello può anche indicare un errore progressivo (ad esempio, la potenziale perdita di risorse).
* =controllo	Evento significativo che influisce sullo stato del server o sulle risorse
* =info	Informazioni generali che delineano l'andamento complessivo delle attività
* =config	Modifica della configurazione o stato
* =dettaglio	Informazioni generali sull'avanzamento dell'attività secondaria
* =fine	Informazioni di traccia - Traccia generale + valori di entrata, uscita e ritorno del metodo
* =più fine	Informazioni di traccia - Traccia dettagliata
* =migliore	Informazioni di traccia - Una traccia più dettagliata che include tutti i dettagli necessari per il debug dei problemi
* =tutti	Tutti gli eventi vengono registrati

## Configurazione del token LTPA

I token LTPA possono essere utilizzati per evitare che un utente fornisca credenziali nome utente e password su ogni richiesta al server mqweb. È possibile configurare il nome del cookie del token LTPA, l'intervallo di scadenza per i token di autenticazione LTPA e configurare se i token LTPA possono essere utilizzati dalle connessioni HTTP, utilizzando il comando **setmqweb**.

## Prima di iniziare

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmweb** e **setmqweb**:

- ▶ **z/OS** Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
- ▶ **Multi** Su tutti gli altri sistemi operativi, è necessario essere un utente privilegiato.
- ▶ **Linux** ▶ **V 9.4.0** Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.

**Nota:** Se si utilizza sia l'autenticazione IBM MQ Console, sia l'autenticazione token con REST API, l'intervallo di scadenza viene condiviso.



### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).



### Attenzione: Linux ▶ V 9.4.0

Prima di immettere i comandi **setmqweb** o **dspmweb** in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente `MQ_OVERRIDE_DATA_PATH` sulla directory di dati IBM MQ Web Server.

## Informazioni su questa attività

Quando gli utenti accedono a IBM MQ Console, viene generato un token LTPA. Se si utilizza l'autenticazione basata su token con REST API, un token LTPA viene generato quando l'utente accede utilizzando la risorsa `/login` REST API con il metodo HTTP POST. Questo token viene restituito in un cookie. Il token viene utilizzato per autenticare l'utente senza che all'utente venga richiesto di collegarsi di nuovo con il relativo ID utente e password, fino alla scadenza del token. L'intervallo di scadenza predefinito è 120 minuti.

Il nome del cookie che include il token LTPA varia in base alla piattaforma:

- ▶ **MQ Appliance** Su IBM MQ Appliance, il token LTPA è `LtpaToken2`. Questo valore non può essere modificato.
- ▶ **z/OS** ▶ **ALW** Per impostazione predefinita, su tutte le altre piattaforme, il nome del cookie che include il token LTPA inizia con `LtpaToken2` e include un suffisso che può cambiare quando viene riavviato il server mqweb. Questo nome cookie casuale consente l'esecuzione di più di un server mqweb sullo stesso sistema. Tuttavia, se si desidera che il nome cookie rimanga un valore congruente, è possibile specificare il nome che il cookie ha utilizzando il comando **setmqweb**.

▶ **z/OS** ▶ **IBM i** ▶ **ALW** Se si abilitano entrambe le porte HTTP e HTTPS, un token LTPA emesso per una richiesta HTTPS può essere riutilizzato per una richiesta HTTP. Questo comportamento è disabilitato per impostazione predefinita, ma è possibile abilitarlo utilizzando il comando **setmqweb**.

## Procedura

- Visualizzare la scadenza corrente del token LTPA, il nome del cookie del token LTPA e se il token LTPA può essere utilizzato per le richieste HTTP utilizzando il seguente comando:

```
dspmweb properties -a
```

- Il campo `ltpaCookieName` mostra il nome del cookie del token LTPA. Se non è stato impostato un nome cookie, il valore di questa proprietà è `LtpaToken2_${env.MQWEB_LTPA_SUFFIX}` on AIX, Linux, and Windows o `LtpaToken2_${httpsPort}` on z/OS, . La variabile dopo il prefisso `LtpaToken2_` viene utilizzata dal server mqweb per generare un nome univoco per il cookie. Non è possibile impostare questa variabile, ma è possibile modificare `ltpaCookieName` in un valore di propria scelta.
- Il campo `ltpaExpiration` mostra la scadenza del token LTPA.
- Il campo `secureLtpa` è impostato su `false` se i token LTPA possono essere utilizzati dalle richieste HTTP.

- Configurare la scadenza del token LTPA:

- Impostare la scadenza del token LTPA immettendo il seguente comando:

```
setmqweb properties -k ltpaExpiration -v time
```

dove *time* specifica l'ora, in minuti, prima della scadenza del token LTPA e della disconnessione dell'utente.

- Reimpostare la scadenza del token LTPA sul valore predefinito di 120 minuti immettendo il seguente comando:

```
setmqweb properties -k ltpaExpiration -d
```

- 

Configurare il nome cookie del token LTPA:

- Impostare il nome del cookie del token LTPA immettendo il seguente comando:

```
setmqweb properties -k ltpaCookieName -v name
```

dove *name* specifica un nome univoco per il cookie del token LTPA.

- Reimpostare il nome del cookie del token LTPA sul valore predefinito, dove un prefisso di `LtpaToken2_` è seguito da caratteri casuali, immettendo il seguente comando:

```
setmqweb properties -k ltpaCookieName -d
```

- 

Configurare se il token LTPA può essere utilizzato dalle connessioni HTTP immettendo il seguente comando:

```
setmqweb properties -k secureLtpa -v secure
```

dove *secure* specifica se il token LTPA può essere utilizzato sia da connessioni HTTP non sicure che da connessioni HTTPS sicure. Il valore `false` consente alle connessioni HTTP e HTTPS di utilizzare lo stesso token LTPA.

## Configurazione del comportamento della connessione del gestore code remoto per IBM MQ Console

Quando si utilizza IBM MQ Console, è possibile creare connessioni ai gestori code remoti. In altre parole, è possibile connettersi a gestori code che non fanno parte della stessa installazione del server mqweb che esegue IBM MQ Console. Esistono diverse opzioni di configurazione che è possibile impostare per controllare il comportamento delle connessioni del gestore code remoto.

## Prima di iniziare

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmweb** e **setmqweb**:

- ▶ **z/OS** Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
- ▶ **Multi** Su tutti gli altri sistemi operativi, è necessario essere un utente privilegiato.
- ▶ **Linux** ▶ **V 9.4.0** Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.



### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).



### Attenzione:

**Linux** ▶ **V 9.4.0**

Prima di immettere i comandi **setmqweb** o **dspmweb** in un'installazione IBM MQ Web Server autonoma, è necessario impostare la variabile di ambiente **`MQ_OVERRIDE_DATA_PATH`** sulla directory di dati IBM MQ Web Server.

## Informazioni su questa attività

È possibile impostare le seguenti opzioni di configurazione:

- Se le connessioni del gestore code remoto sono consentite.
- Indica se è possibile aggiungere le connessioni utilizzando IBM MQ Console solo la riga comandi.
- Indica se i gestori code locali vengono visualizzati in IBM MQ Console quando sono consentite le connessioni al gestore code remoto.
- Indica se le connessioni del gestore code remoto vengono stabilite automaticamente all'avvio di IBM MQ Console o in caso di errore di connessione.
- L'intervallo di tempo tra ciascun aggiornamento dell'elenco di gestori code remoti visualizzato in IBM MQ Console.

## Procedura

- Per visualizzare le impostazioni di configurazione della connessione del gestore code remoto corrente, immettere il seguente comando:

```
dspmweb properties -a
```



- Il campo `mqConsoleRemoteSupportEnabled` indica se le connessioni al gestore code remoto sono consentite.
- Il campo `mqConsoleRemoteUIAdmin` indica se è possibile aggiungere connessioni al gestore code remoto utilizzando IBM MQ Console.
- Il campo `mqConsoleRemoteAllowLocal` indica se vengono visualizzati i gestori code locali.

– Il campo `mqConsoleRemotePollTime` indica il numero di secondi tra ogni aggiornamento dell'elenco di gestori code remoti.

- Per impedire o consentire le connessioni del gestore code remoto con IBM MQ Console, immettere il seguente comando:

```
setmqweb properties -k mqConsoleRemoteSupportEnabled -v true or false
```

dove `true` consente le connessioni del gestore code remoto o `false` impedisce le connessioni del gestore code remoto.

**Nota:**   Se il server `mqweb` viene eseguito in un'installazione autonoma di IBM MQ Web Server, la proprietà `mqConsoleRemoteSupportEnabled` non è valida. Il IBM MQ Web Server autonomo supporta le connessioni solo ai gestori code remoti.

- Per impedire o consentire l'aggiunta di connessioni al gestore code remoto utilizzando IBM MQ Console solo dalla riga comandi, immettere il seguente comando:



```
setmqweb properties -k mqConsoleRemoteUIAdmin -v true or false
```

dove `true` consente di aggiungere connessioni al gestore code remoto utilizzando IBM MQ Console e la riga comandi oppure `false` consente di aggiungere connessioni al gestore code remoto solo utilizzando il comando `setmqweb remote` sulla riga comandi.

- Per impedire o consentire la visualizzazione dei gestori code locali in IBM MQ Console quando sono consentite le connessioni al gestore code remoto, immettere il seguente comando:

```
setmqweb properties -k mqConsoleRemoteAllowLocal -v true or false
```

dove `true` consente ai gestori code locali di essere visualizzati oppure `false` nasconde i gestori code locali.

**Nota:**   Se il server `mqweb` viene eseguito in un'installazione autonoma di IBM MQ Web Server, la proprietà `mqConsoleRemoteAllowLocal` non è valida. Il IBM MQ Web Server autonomo supporta le connessioni solo ai gestori code remoti.

- Per impostare il periodo di tempo tra ogni aggiornamento dell'elenco di gestori code remoti visualizzato in IBM MQ Console, immettere il seguente comando:

```
setmqweb properties -k mqConsoleRemotePollTime -v seconds
```

dove `seconds` è impostato su un valore intero del numero di secondi tra ogni aggiornamento dell'elenco di gestori code remoti.



#### Riferimenti correlati

 [pid setmqweb](#)  
[dspmqweb](#)

## Configurazione del gateway amministrative REST API

Quando il gateway amministrative REST API è abilitato, è possibile eseguire la gestione remota con il REST API utilizzando un gestore code gateway. È possibile configurare il gestore code utilizzato come gestore code gateway predefinito oppure è possibile impedire la gestione remota disabilitando il gateway amministrative REST API utilizzando il comando `setmqweb`.

### Prima di iniziare

**Nota:**   Se il server `mqweb` viene eseguito in un'installazione IBM MQ Web Server autonoma, questa attività non è applicabile. amministrative REST API non è disponibile in un'installazione IBM MQ Web Server autonoma.

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 **dspmweb** 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:

Prima di emettere i comandi **setmqweb** o **dspmweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).

## Informazioni su questa attività

Quando il server mqweb viene eseguito in un'installazione IBM MQ, il gateway amministrativo REST API è abilitato per impostazione predefinita.

Il gestore code del gateway predefinito viene utilizzato quando si verificano entrambe le seguenti condizioni:

- Un gestore code non è stato specificato nell'installazione `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.

Per ulteriori informazioni sulla gestione remota con REST API, consultare [Gestione remota utilizzando REST API](#).

## Procedura

- Visualizzare la configurazione corrente del gateway amministrativo REST API utilizzando il comando seguente:

```
dspmweb properties -a
```

Il campo `mqRestGatewayEnabled` mostra se il gateway è abilitato e il campo `mqRestGatewayQmgr` mostra il nome del gestore code del gateway predefinito.

- Configurare se il gateway amministrativo REST API è abilitato utilizzando il seguente comando:

```
setmqweb properties -k mqRestGatewayEnabled -v enabled
```

dove `enabled` è il valore **true** per abilitare il gateway amministrativo REST API o **false** in caso contrario.

- Configurare il gestore code utilizzato come gestore code del gateway predefinito:

- Impostare il gestore code del gateway predefinito utilizzando il seguente comando:

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

dove `qmgrName` è il nome di un gestore code nella stessa installazione del server mqweb.

- Annullare l'impostazione del gestore code gateway predefinito utilizzando il seguente comando:


```
setmqweb properties -k mqRestGatewayQmgr -d
```



## Configurazione di messaging REST API

È possibile configurare messaging REST API in diversi modi. È possibile scegliere di abilitare o disabilitare la funzione messaging REST API. È possibile scegliere il numero massimo di connessioni in pool che possono essere utilizzate da messaging REST API e il comportamento di messaging REST API quando tutte le connessioni sono in uso. È inoltre possibile scegliere il contesto utente utilizzato per l'autorizzazione quando si utilizza messaging REST API per inviare, ricevere, sfogliare o pubblicare un messaggio.

### Procedura





- [“Abilitazione di messaging REST API” a pagina 857](#)
- [“Configurazione del pool di connessioni per messaging REST API” a pagina 858](#)
-  [“Configurazione del contesto utente utilizzato per l'autorizzazione in messaging REST API” a pagina 861](#)

### Abilitazione di messaging REST API

È possibile configurare se il messaging REST API è abilitato utilizzando il comando **setmqweb**. Per impostazione predefinita, messaging REST API è abilitato.

### Prima di iniziare

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmqweb** e **setmqweb**:

-  Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmqweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
-  Su tutti gli altri sistemi operativi, è necessario essere un [utente privilegiato](#).
-   Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.



#### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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

- Visualizzare la configurazione attuale di messaging REST API utilizzando il seguente comando:

```
dspmqweb properties -a
```

Il campo `mqRestMessagingEnabled` mostra se il messaging REST API è abilitato. Se il valore è `True`, messaging REST API è abilitato.

- Abilitare messaging REST API utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingEnabled -v true
```

- Disabilitare messaging REST API utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingEnabled -v false
```

### Attività correlate

“Configurazione del pool di connessioni per messaging REST API” a pagina 858

È possibile configurare il numero massimo di connessioni in pool che possono essere utilizzate da messaging REST API e il comportamento di messaging REST API quando tutte le connessioni sono in uso.

“Configurazione del contesto utente utilizzato per l'autorizzazione in messaging REST API” a pagina 861

**V 9.4.0** È possibile configurare quale contesto utente viene utilizzato per l'autorizzazione quando si utilizza messaging REST API per inviare, ricevere, sfogliare o pubblicare un messaggio. In altre parole, è possibile scegliere se l'utente collegato a messaging REST API o l'utente che ha avviato il server mqweb, viene utilizzato per l'autorizzazione.

“Configurazione della modalità di connessione per messaging REST API” a pagina 860

È possibile configurare messaging REST API per la connessione a gestori code locali o remoti.

## Configurazione del pool di connessioni per messaging REST API

È possibile configurare il numero massimo di connessioni in pool che possono essere utilizzate da messaging REST API e il comportamento di messaging REST API quando tutte le connessioni sono in uso.

### Prima di iniziare

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.
- **Linux** **V 9.4.0** Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.



#### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).

## Informazioni su questa attività

Per ottimizzare le prestazioni di messaging REST API, le connessioni ai gestori code IBM MQ vengono raggruppate in pool. Ciò significa che, invece di ogni richiesta REST che crea, utilizza ed elimina la propria connessione, ogni richiesta REST utilizza una connessione da un pool di connessione. Per impostazione predefinita, sono disponibili 20 connessioni per ogni lotto di gestori code ed è possibile scegliere tra tre opzioni per la gestione delle richieste quando tutte le connessioni sono in uso:

- messaging REST API può creare una nuova connessione non in pool da utilizzare per la richiesta. Questo è il comportamento predefinito.
- messaging REST API può restituire un errore.
- messaging REST API può attendere che una connessione in pool diventi disponibile. Questa attesa è un'attesa indefinita.

È possibile modificare il numero massimo di connessioni in pool e il comportamento predefinito di messaging REST API quando tutte le connessioni vengono utilizzate utilizzando il comando **setmqweb properties**.

## Procedura

- Visualizzare la configurazione corrente utilizzando il seguente comando:

```
dspmqweb properties -a
```

- Il campo `mqRestMessagingFullPoolBehavior` mostra il funzionamento di messaging REST API quando tutte le connessioni nel lotto sono in uso. Se il valore è `block`, messaging REST API deve attendere che una connessione diventi disponibile. Se il valore è `error`, messaging REST API deve restituire un errore. Se il valore è `overflow`, messaging REST API deve creare una connessione non in pool da utilizzare ed eliminare la connessione dopo l'utilizzo.
- Il campo `mqRestMessagingMaxPoolSize` mostra la dimensione massima del lotto connessioni.
- Configurare il funzionamento di messaging REST API quando tutte le connessioni nel lotto sono in uso utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingFullPoolBehavior -v action
```

dove *action* specifica l'azione da intraprendere. *action* può essere uno dei seguenti valori:

### bitmap

Quando tutte le connessioni nel pool sono in uso, attendere che una connessione diventi disponibile.

### errore

Quando tutte le connessioni nel pool sono in uso, restituire un errore.

### in eccesso

Quando tutte le connessioni nel pool sono in uso, creare una connessione non in pool da utilizzare ed eliminare la connessione dopo che è stata utilizzata.

- Configurare la dimensione massima del lotto connessioni per ciascun lotto gestori code utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingMaxPoolSize -v size
```

dove *size* specifica la dimensione del pool.

**Nota:** Se è impostato un valore elevato per `mqRestMessagingMaxPoolSize` e molti gestori code sono connessi, aumentare la dimensione massima dell'heap del server mqweb. Per ulteriori informazioni, consultare [ottimizzazione della JVM del server mqweb](#).

## Attività correlate

[“Abilitazione di messaging REST API” a pagina 857](#)

È possibile configurare se il messaging REST API è abilitato utilizzando il comando **setmqweb**. Per impostazione predefinita, messaging REST API è abilitato.

“Configurazione del contesto utente utilizzato per l'autorizzazione in messaging REST API” a pagina 861

**V 9.4.0** È possibile configurare quale contesto utente viene utilizzato per l'autorizzazione quando si utilizza messaging REST API per inviare, ricevere, sfogliare o pubblicare un messaggio. In altre parole, è possibile scegliere se l'utente collegato a messaging REST API o l'utente che ha avviato il server mqweb, viene utilizzato per l'autorizzazione.

“Configurazione della modalità di connessione per messaging REST API” a pagina 860

È possibile configurare messaging REST API per la connessione a gestori code locali o remoti.

## **V 9.4.0** Configurazione della modalità di connessione per messaging REST API

È possibile configurare messaging REST API per la connessione a gestori code locali o remoti.

### Prima di iniziare

**Nota:** **V 9.4.0** **Linux** Se il server mqweb viene eseguito in un'installazione autonoma di IBM MQ Web Server, questa attività non è applicabile. Il IBM MQ Web Server autonomo supporta le connessioni solo ai gestori code remoti.

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:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).

### Informazioni su questa attività

La modalità di collegamento predefinita per messaging REST API varia in base al tipo di installazione che esegue il server mqweb:

- In una installazione di IBM MQ, per default messaging REST API si connette solo ai gestori code locali nella stessa installazione del server mqweb. Completare i passi in questa attività per visualizzare e modificare la configurazione della connessione.
- **V 9.4.0** **Linux** In un'installazione IBM MQ Web Server autonoma, messaging REST API supporta le connessioni solo ai gestori code remoti. La configurazione della connessione non può essere visualizzata o modificata.

## Procedura

- Visualizzare la configurazione attuale di messaging REST API utilizzando il seguente comando:

```
dspmweb properties -a
```

Il campo `mqRestMessagingConnectionMode` mostra la modalità di connessione corrente. Se il valore è `local`, il messaging REST API può connettersi solo ai gestori code nella stessa installazione del server mqweb. Se il valore è `remote`, il messaging REST API può connettersi ai gestori code remoti.

- Configurare il server mqweb per consentire a messaging REST API di connettersi solo ai gestori code che si trovano nella stessa installazione del server mqweb utilizzando i seguenti comandi:

```
setmqweb properties -k mqRestMessagingConnectionMode -v local
endmqweb
strmqweb
```

- Configurare il server mqweb per consentire al messaging REST API di collegarsi ai gestori code remoti utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingConnectionMode -v remote
endmqweb
strmqweb
```

## Operazioni successive

Se si configura il server mqweb per consentire a messaging REST API di connettersi ai gestori code remoti, è necessario fornire le informazioni di connessione per ciascun gestore code a cui si desidera connettersi. Per ulteriori informazioni su come fornire le informazioni di connessione, fare riferimento a [Impostazione di un gestore code remoto da utilizzare con messaging REST API](#).

### Attività correlate

[“Abilitazione di messaging REST API” a pagina 857](#)

È possibile configurare se il messaging REST API è abilitato utilizzando il comando **setmqweb**. Per impostazione predefinita, messaging REST API è abilitato.

[“Configurazione del pool di connessioni per messaging REST API” a pagina 858](#)

È possibile configurare il numero massimo di connessioni in pool che possono essere utilizzate da messaging REST API e il comportamento di messaging REST API quando tutte le connessioni sono in uso.

[“Configurazione del contesto utente utilizzato per l'autorizzazione in messaging REST API” a pagina 861](#)

**V 9.4.0** È possibile configurare quale contesto utente viene utilizzato per l'autorizzazione quando si utilizza messaging REST API per inviare, ricevere, sfogliare o pubblicare un messaggio. In altre parole, è possibile scegliere se l'utente collegato a messaging REST API o l'utente che ha avviato il server mqweb, viene utilizzato per l'autorizzazione.

### **V 9.4.0** Configurazione del contesto utente utilizzato per l'autorizzazione in messaging REST API

**V 9.4.0** È possibile configurare quale contesto utente viene utilizzato per l'autorizzazione quando si utilizza messaging REST API per inviare, ricevere, sfogliare o pubblicare un messaggio. In altre parole, è possibile scegliere se l'utente collegato a messaging REST API o l'utente che ha avviato il server mqweb, viene utilizzato per l'autorizzazione.

## Prima di iniziare

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmweb** e **setmqweb**:

- ▶ **z/OS** Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
- ▶ **Multi** Su tutti gli altri sistemi operativi, è necessario essere un utente privilegiato.
- ▶ **Linux** ▶ **V 9.4.0** Se il server mqweb fa parte di un'installazione autonoma di IBM MQ Web Server, è necessario disporre dell'accesso in scrittura al file `mqwebuser.xml` nella directory di dati IBM MQ Web Server.



#### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).

## Informazioni su questa attività

- Quando l'ID utente utilizzato è l'ID utente collegato a messaging REST API, **MQMD.UserIDentifier** è impostato sull'ID utente collegato all'API REST. **MQMD.AppIdentityData** è impostato sull'ID utente collegato all'API REST.
- Quando l'ID utente utilizzato è l'ID utente che ha avviato il server mqweb, il campo **MQMD.UserIDentifier** viene lasciato vuoto. **MQMD.AppIdentityData** è impostato sull'ID utente collegato all'API REST.

Consultare [MQMD](#) per ulteriori informazioni sulle parti del descrittore del messaggio IBM MQ.

## Procedura

- Visualizzare la configurazione attuale di messaging REST API utilizzando il seguente comando:

```
dspmweb properties -a
```

Il campo `mqRestMessagingAdoptWebUserContext` mostra l'ID utente utilizzato per l'autorizzazione quando si inviano, pubblicano, ricevono o sfogliano i messaggi. Se il valore è `True`, l'utente collegato a messaging REST API viene utilizzato per l'autorizzazione. Se il valore è `False`, l'utente che ha avviato il server mqweb viene utilizzato per l'autorizzazione.

- Configurare messaging REST API per utilizzare l'ID utente dell'utente collegato a messaging REST API per l'autorizzazione utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingAdoptWebUserContext -v true
```

Quando `mqRestMessagingAdoptWebUserContext` è impostato su `true`, **MQMD.UserIDentifier** è impostato sull'ID utente collegato all'API REST. **MQMD.AppIdentityData** è impostato sull'ID utente collegato all'API REST.

- Configurare messaging REST API per utilizzare l'ID utente dell'utente che ha avviato il server mqweb utilizzando il seguente comando:

```
setmqweb properties -k mqRestMessagingAdoptWebUserContext -v false
```

Quando **mqRestMessagingAdoptWebUserContext** è impostata su `false`, **MQMD.UserIdentifier** viene lasciato vuoto. **MQMD.AppIdentityData** è impostato sull'ID utente collegato all'API REST.

### Attività correlate

[“Abilitazione di messaging REST API” a pagina 857](#)

È possibile configurare se il messaging REST API è abilitato utilizzando il comando **setmqweb**. Per impostazione predefinita, messaging REST API è abilitato.

[“Configurazione del pool di connessioni per messaging REST API” a pagina 858](#)

È possibile configurare il numero massimo di connessioni in pool che possono essere utilizzate da messaging REST API e il comportamento di messaging REST API quando tutte le connessioni sono in uso.

[“Configurazione della modalità di connessione per messaging REST API” a pagina 860](#)

È possibile configurare messaging REST API per la connessione a gestori code locali o remoti.

## Configurazione di REST API per MFT

Per impostazione predefinita, REST API per MFT non è abilitato. È possibile configurare se REST API per MFT è abilitato, impostare il gestore code di coordinamento, impostare il gestore code di comandi e specificare il timeout di riconnessione di MFT utilizzando il comando **setmqweb properties**.



### Procedura

- [“Abilitazione di REST API per MFT” a pagina 863](#)
- [“Configurazione del gestore code di coordinamento per REST API per MFT” a pagina 864](#)
- [“Configurazione del gestore code comandi per REST API per MFT” a pagina 865](#)
- [“Configurazione dei valori di timeout di REST API per MFT” a pagina 866](#)



## Abilitazione di REST API per MFT

Prima di poter utilizzare REST API per MFT, è necessario abilitare REST API per MFT. È possibile configurare se il REST API per MFT è abilitato utilizzando il comando **setmqweb**. Per impostazione predefinita, REST API per MFT non è abilitato.

### Prima di iniziare

**Nota:**   Se il server mqweb viene eseguito in un'installazione autonoma di IBM MQ Web Server, questa attività non è applicabile. REST API per MFT non è disponibile in una installazione IBM MQ Web Server autonoma.

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmqweb** e **setmqweb**:

-  Su z/OS, è necessario disporre dell'autorità per eseguire i comandi **dspmqweb** e **setmqweb** e l'accesso in scrittura al file `mqwebuser.xml`.
-  Su tutti gli altri sistemi operativi, è necessario essere un [utente privilegiato](#).



#### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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

1. Visualizza la configurazione corrente di REST API per MFT utilizzando il seguente comando:

```
dspmweb properties -a
```

Il campo `mqRestMftEnabled` mostra se REST API per MFT è abilitato. Il valore è `True` se REST API per MFT è abilitato o `False` in caso contrario.

2. Abilitare o disabilitare REST API per MFT utilizzando uno dei comandi riportati di seguito:

- Abilitare REST API per MFT utilizzando il seguente comando:

```
setmqweb properties -k mqRestMftEnabled -v true
```

- Disabilita REST API per MFT utilizzando il seguente comando:

```
setmqweb properties -k mqRestMftEnabled -v false
```

3. Riavviare il server mqweb immettendo i seguenti comandi:

```
endmqweb
strmqweb
```



## Operazioni successive

Se è stato abilitato REST API per MFT, è necessario impostare il nome del gestore code di coordinamento prima di poter utilizzare REST API per MFT. Per ulteriori informazioni su come impostare il gestore code di coordinamento, consultare [“Configurazione del gestore code di coordinamento per REST API per MFT”](#) a pagina 864.



## Configurazione del gestore code di coordinamento per REST API per MFT

Prima di poter utilizzare REST API per MFT, è necessario configurare un gestore code che funga da gestore code di coordinamento per le transazioni MFT. È possibile impostare il gestore code di coordinamento utilizzando il comando **setmqweb**.

## Prima di iniziare

**Nota:**   Se il server mqweb viene eseguito in un'installazione autonoma di IBM MQ Web Server, questa attività non è applicabile. REST API per MFT non è disponibile in una installazione IBM MQ Web Server autonoma.

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmweb** e **setmqweb**:

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



### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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

1. Visualizza la configurazione corrente di REST API per MFT utilizzando il seguente comando:

```
dspmweb properties -a
```

Il campo `mqRestMftCoordinationQmgr` mostra il nome del gestore code di coordinamento.

2. Configurare il gestore code di coordinamento utilizzando il seguente comando:

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

dove *qmgrName* è il nome del gestore code di coordinamento. Il gestore code di coordinamento deve trovarsi sulla macchina su cui è in esecuzione il server mqweb. Per default, questo nome del gestore code è vuoto. Se un valore non è impostato, il REST API per MFT non funziona.

3. Riavviare il server mqweb immettendo i seguenti comandi:

```
endmqweb
strmqweb
```



## Operazioni successive

- Verificare che REST API per MFT sia abilitato. Per ulteriori informazioni, consultare [“Abilitazione di REST API per MFT”](#) a pagina 863.
- Se si desidera utilizzare REST API per MFT per inoltrare le richieste di creazione, è necessario impostare il nome del gestore code comandi. Ad esempio, se si desidera utilizzare un comando REST API come **create transfer**, è necessario impostare il nome del gestore code comandi. Per ulteriori informazioni, fare riferimento a [“Configurazione del gestore code comandi per REST API per MFT”](#) a pagina 865.
- È possibile configurare i valori di timeout REST API per MFT . Il timeout predefinito è 30 minuti. Per ulteriori informazioni, consultare [“Configurazione dei valori di timeout di REST API per MFT”](#) a pagina 866.
- Per utilizzare REST API per MFT, un utente deve essere autenticato sul server mqweb e deve essere membro di uno o più ruoli `MFTWebAdmin` o `MFTWebAdminRO` . Per ulteriori informazioni sulla configurazione degli utenti, consultare [Configurazione di utenti e ruoli per REST API](#).

## Configurazione del gestore code comandi per REST API per MFT

Prima di poter utilizzare REST API per MFT per inoltrare richieste di creazione, è necessario impostare il nome del gestore code comandi. Ad esempio, per utilizzare la risorsa **create transfer** , è necessario impostare il nome del gestore code comandi. È possibile impostare il nome gestore code comandi utilizzando il comando **setmqweb** .

## Prima di iniziare

**Nota:**   Se il server mqweb viene eseguito in un'installazione autonoma di IBM MQ Web Server , questa attività non è applicabile. REST API per MFT non è disponibile in una installazione IBM MQ Web Server autonoma.

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmweb** 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:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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

1. Visualizza la configurazione corrente di REST API per MFT utilizzando il seguente comando:

```
dspmqweb properties -a
```

Il campo `mqRestMftCommandQmgr` mostra il nome del gestore code comandi.

2. Configurare il gestore code comandi utilizzando il comando seguente:

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

dove `qmgrName` è il nome del gestore code comandi. Il gestore code comandi deve trovarsi sulla macchina su cui è in esecuzione il server mqweb. Per default, questo nome del gestore code è vuoto. Se un valore non è impostato, REST API per MFT per un comando di creazione non funziona.

3. Riavviare il server mqweb immettendo i seguenti comandi:

```
endmqweb
strmqweb
```



## Operazioni successive

- Verificare che REST API per MFT sia abilitato. Per ulteriori informazioni, consultare [“Abilitazione di REST API per MFT”](#) a pagina 863.
- Verificare che sia impostato un gestore code di coordinamento. Per ulteriori informazioni, consultare [“Configurazione del gestore code di coordinamento per REST API per MFT”](#) a pagina 864.
- È possibile configurare i valori di timeout REST API per MFT. Il timeout predefinito è 30 minuti. Per ulteriori informazioni, consultare [“Configurazione dei valori di timeout di REST API per MFT”](#) a pagina 866.
- Per utilizzare REST API per MFT, un utente deve essere autenticato sul server mqweb e deve essere membro di uno o più ruoli `MFTWebAdmin` o `MFTWebAdminRO`. Per ulteriori informazioni sulla configurazione degli utenti, consultare [Configurazione di utenti e ruoli per REST API](#).



## Configurazione dei valori di timeout di REST API per MFT

È possibile configurare l'intervallo di tempo, in minuti, dopo il quale REST API per MFT interrompe il tentativo di connessione al gestore code di coordinamento dopo che la connessione è stata interrotta. Il timeout predefinito è 30 minuti. È possibile configurare questo timeout utilizzando il comando **setmqweb**.

## Prima di iniziare

**Nota:**   Se il server mqweb viene eseguito in un'installazione IBM MQ Web Server autonoma, questa attività non è applicabile. REST API per MFT non è disponibile in una installazione IBM MQ Web Server autonoma.

Per completare questa attività, è necessario essere un utente con determinati privilegi in modo da poter utilizzare i comandi **dspmqweb** e **setmqweb**:

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



### Attenzione:

Prima di emettere i comandi **setmqweb** o **dspmqweb** su z/OS, è necessario impostare la variabile di ambiente `WLP_USER_DIR` in modo che la variabile punti alla propria configurazione del server mqweb.

Per impostare la variabile di ambiente `WLP_USER_DIR`, 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](#).

## Informazioni su questa attività

È possibile configurare il timeout per REST API per MFT.

REST API per MFT tenta di ristabilire la connessione immediatamente dopo che la connessione al gestore code di coordinamento è stata interrotta. Se questo tentativo ha esito negativo, c'è un intervallo di cinque minuti tra ogni tentativo di riconnessione, fino a quando il timeout non è trascorso. Pertanto, l'impostazione di un valore compreso tra 0 e 5 determina un solo tentativo di riconnessione.

Dopo il timeout della riconnessione, viene effettuato il successivo tentativo di riconnessione quando viene richiamata una delle risorse REST API per MFT. Se questo tentativo di riconnessione ha esito negativo, MFT tenta nuovamente di riconnettersi ogni cinque minuti fino a quando non è trascorso il timeout di riconnessione.

## Procedura

1. Visualizza la configurazione corrente di REST API per MFT utilizzando il seguente comando:

```
dspmqweb properties -a
```

Il campo `mqRestMftReconnectTimeoutInMinutes` mostra il valore di timeout della riconnessione, fino a quando i servizi REST di trasferimento MFT non smettono di tentare la connessione al gestore code di coordinamento.

2. Configurare il timeout, in minuti, dopo il quale REST API per MFT interrompe il tentativo di connessione al gestore code di coordinamento:

- Reimpostare il timeout sul valore predefinito di 30 minuti:

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -d
```

- Impostare il timeout:

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
```

dove *time* specifica il tempo, in minuti, prima che si verifichi il timeout.

Se questo valore è impostato tra 0-5, REST API per MFT tenta di riconnettersi al gestore code di coordinamento solo una volta. Se la connessione non riesce, non ci sono tentativi di ristabilire la connessione fino a quando non viene richiamato REST API .

Se questo valore è impostato su -1, REST API per MFT tenta di riconnettersi fino a quando la connessione non viene eseguita correttamente.

3. Riavviare il server mqweb immettendo i seguenti comandi:

```
endmqweb
startmqweb
```

## Operazioni successive

- Verificare che REST API per MFT sia abilitato. Per ulteriori informazioni, consultare [“Abilitazione di REST API per MFT”](#) a pagina 863.
- Verificare che sia impostato un gestore code di coordinamento. Per ulteriori informazioni, consultare [“Configurazione del gestore code di coordinamento per REST API per MFT”](#) a pagina 864.
- Se si desidera utilizzare REST API per MFT per inoltrare le richieste di creazione, è necessario impostare il nome del gestore code comandi. Ad esempio, se si desidera utilizzare un comando REST API come **create transfer**, è necessario impostare il nome del gestore code comandi. Per ulteriori informazioni, fare riferimento a [“Configurazione del gestore code comandi per REST API per MFT”](#) a pagina 865.
- Per utilizzare REST API per MFT, un utente deve essere autenticato sul server mqweb e deve essere membro di uno o più ruoli MFTWebAdmin o MFTWebAdminRO . Per ulteriori informazioni sulla configurazione degli utenti, consultare [Configurazione di utenti e ruoli per REST API](#).

## Ottimizzazione della JVM del server mqweb

Per impostazione predefinita, la JVM (mqweb server Java Virtual Machine ) utilizza i valori predefiniti specifici della piattaforma per i parametri di configurazione, ad esempio la dimensione minima e massima dell'heap e la dimensione della cache delle classi.

### Informazioni su questa attività

Potrebbe essere necessario modificare i valori predefiniti per migliorare le prestazioni o per risolvere i problemi. Ad esempio, se `java.lang.OutOfMemoryError` viene generato dal server mqweb, è necessario aumentare la dimensione massima dell'heap. È inoltre necessario aumentare la dimensione dell'heap se si sta tentando di caricare un numero elevato di oggetti coda.


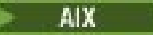
Se si verificano problemi con la visualizzazione delle informazioni di configurazione del dashboard in IBM MQ Console, è necessario impostare una variabile che determina la codifica file della configurazione. È possibile modificare i valori predefiniti nel file `jvm.options` .


### Procedura


1. Aprire il file `jvm.options`.


Il file `jvm.options` può essere trovato in una delle seguenti directory:

- In un'installazione IBM MQ :



–   Su AIX o Linux: `/var/mqm/web/installations/  
installationName/servers/mqweb`

–  Su Windows:  
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`, dove  
`MQ_DATA_PATH` è il percorso dati IBM MQ . Questo percorso è il percorso dati selezionato durante  
l'installazione di IBM MQ. Per impostazione predefinita, questo è `C:\ProgramData\IBM\MQ`.

–  Su IBM i: `MQ_DATA_PATH/web/installations/Installation1/`

–  Su z/OS: `WLP_user_directory/servers/mqweb`

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

-   In un'installazione autonoma di IBM MQ Web Server :  
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST/servers/mqweb`

dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente `MQ_OVERRIDE_DATA_PATH`.

2. Opzionale: Impostare la dimensione heap massima aggiungendo la seguente linea al file:

```
-XmxMaxSizem
```

Dove `MaxSize` specifica la dimensione massima dell'heap, in MB.

Ad esempio, la seguente riga imposta la dimensione heap massima su 1GB:

```
-Xmx1024m
```

3. Opzionale: Impostare la dimensione heap minima aggiungendo la riga seguente al file:

```
-XmsMinSizem
```

Dove `MinSize` specifica la dimensione minima dell'heap, in MB. L'aumento della dimensione heap minima dal valore predefinito può ridurre il tempo impiegato per avviare il server mqweb.

Ad esempio, la seguente riga imposta la dimensione heap minima su 512MB:

```
-Xms512m
```

4. Opzionale: Impostare la dimensione della cache di classi aggiungendo la seguente riga al file:

```
-XscmxSizem
```

Dove `Dimensione` specifica la dimensione della cache delle classi, in MB.


Ad esempio, la seguente riga imposta la dimensione della cache di classi su 100MB: :

```
-Xscmx100m
```

La cache delle classi condivisa Java è utilizzata per memorizzare i dati come le classi caricate e il codice AOT (Ahead - Of - Time) compilato.

La cache delle classi riduce significativamente il tempo impiegato per avviare il server mqweb. La prima volta che il server mqweb viene avviato, viene creata la cache delle classi e il server può impiegare un tempo significativo per l'avvio. I riavvii successivi del server saranno molto più rapidi poiché le classi possono essere caricate dalla cache di classi condivisa.

L'aumento della dimensione della cache delle classi rispetto al valore predefinito può ridurre il tempo impiegato per avviare il server mqweb.

 La cache delle classi viene ricreata quando il server mqweb viene avviato su un altro sistema z/OS . Pertanto, l'avvio del server mqweb su un altro sistema z/OS in un sysplex può richiedere molto più tempo rispetto al riavvio del server sullo stesso sistema.

Notare che le modifiche a questo valore diventano effettive solo quando viene creata la cache delle classi. La cache delle classi viene creata quando il server mqweb viene avviato per la prima volta o dopo che la cache delle classi è stata eliminata utilizzando il programma di utilità della cache delle classi Java .

5. Richiesto: Verificare che il file contenga le seguenti righe per specificare la codifica file utilizzata quando REST API elabora i dati e per le informazioni di configurazione del pannello di controllo utente in IBM MQ Console:

```
-Dfile.encoding=UTF-8
-Ddefault.client.encoding=UTF-8
```

6. Riavviare il server mqweb.

**z/OS** Su z/OS, arrestare e riavviare l'attività avviata del server mqweb.

**Multi** Su tutte le altre piattaforme, immettere i seguenti comandi sulla riga comandi:

```
endmqweb
strmqweb
```

## Struttura del file del componente di installazione IBM MQ Console e REST API

Ci sono due serie di strutture di directory associate al componente di installazione IBM MQ Console e REST API . Una struttura di directory contiene file che è possibile modificare. L'altra struttura di directory contiene file che non possono essere modificati.

### File modificabili

I file modificabili dall'utente sono disposti come parte dell'installazione iniziale del componente di installazione IBM MQ Console e REST API . Poiché questi file possono essere modificati, non vengono modificati quando viene applicata la manutenzione.

L'ubicazione dei file modificabili dell'utente dipende dal sistema operativo e dal prodotto installato.

• In un'installazione IBM MQ , i file modificabili dall'utente si trovano in una delle seguenti directory:

- **Linux** **AIX** Su AIX o Linux: `/var/mqm/web/installations/installationName`
- **Windows** Su Windows: `MQ_DATA_PATH\web\installations\installationName`, dove `MQ_DATA_PATH` è il percorso dati IBM MQ . Questo percorso è il percorso dati selezionato durante l'installazione di IBM MQ. Per impostazione predefinita, questo è `C:\ProgramData\IBM\MQ`.
- **z/OS** Su z/OS: la directory specificata quando lo script `crtmqweb` è stato eseguito per creare la definizione del server mqweb.

• **V 9.4.0** **Linux** In un'installazione autonoma di IBM MQ Web Server :  
`MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST`

dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente `MQ_OVERRIDE_DATA_PATH` .

In questa directory di primo livello, sono presenti le seguenti directory e file:

Directory e file	Descrizione
<code>angular.persistence/</code>	Directory in cui è memorizzata la configurazione del dashboard IBM MQ Console .
<code>servers/</code>	Directory dei server WebSphere Liberty .
<code>servers/mqweb</code>	Directory che contiene la struttura di directory del server mqweb.
<code>servers/mqweb/logs</code>	Directory che contiene i log per il server mqweb.
<code>servers/mqweb/logs/console.log</code>	Log dello stato del server di base e dei messaggi operativi.
<code>servers/mqweb/logs/ffdc</code>	Directory di output FFDC (First Failure Data Capture).

Directory e file	Descrizione
servers/mqweb/logs/messages.log	Log dei messaggi di runtime dal server mqweb, inclusi IBM MQ Console e REST API. I messaggi più vecchi vengono memorizzati in file denominati <code>messages_timestamp.log</code> .
servers/mqweb/logs/trace.log	Log di traccia dal server mqweb, inclusi IBM MQ Console e REST API. La traccia precedente è memorizzata in file denominati <code>trace_timestamp.log</code> . Questi file esistono solo se la traccia è abilitata.
servers/mqweb/logs/state	Stato specifico del server.
servers/mqweb/server.xml	File di configurazione del server principale. Questo file è di sola lettura. Modificare il file <code>mqwebuser.xml</code> per sovrascrivere la configurazione predefinita.
servers/mqweb/mqwebuser.xml	File di configurazione per IBM MQ Console e REST API. Le impostazioni configurate in questo file sovrascrivono la configurazione predefinita. È necessario essere un <u>utente privilegiato</u> per modificare questo file.
servers/mqweb/resources	Directory che contiene varie risorse del server come i keystore.
servers/mqweb/workarea	Directory creata dal server come funziona. Questa directory viene creata dopo la prima esecuzione del server.


## File non modificabili

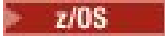
I file non modificabili sono stabiliti come parte dell'installazione iniziale del componente di installazione IBM MQ Console e REST API . Questi file vengono aggiornati quando viene applicata la manutenzione.

L'ubicazione dei file non modificabili dipende dal sistema operativo e dal prodotto installato.

- In un'installazione IBM MQ , i file non modificabili si trovano in una delle seguenti directory:

-  Su AIX, Linux, and Windows: `MQ_INSTALLATION_PATH/web`

-  Su IBM i: `MQ_INSTALLATION_PATH/web`

-  Su z/OS: `installation_directory/web/`

dove `directory_installazione` è il percorso di installazione di IBM MQ for z/OS UNIX System Services Components .

-   In un'installazione IBM MQ Web Server autonoma, la directory in cui è stato decompresso il file di installazione IBM MQ Web Server .

I seguenti file e struttura di directory sono presenti in questa ubicazione:

Directory e file	Descrizione
bin/	Directory che contiene i comandi WebSphere Liberty .

Directory e file	Descrizione
	È necessario essere un utente <u>privilegiato</u> per eseguire gli script in questa directory.
mq/	Struttura di directory che contiene varie risorse IBM MQ .
mq/apps/	Directory che contiene le applicazioni IBM MQ Console e REST API .
mq/etc/	
mq/etc/mqweb.xml	File di configurazione di sola lettura per il server mqweb. Modificare il file mqwebuser.xml per apportare modifiche alla configurazione.
mq/libs	Directory che contiene le librerie condivise per l'utilizzo da parte di IBM MQ Console e REST API.
mq/samp	Directory che contiene esempi.
mq/samp/configuration	Directory che contiene file di configurazione di esempio che possono essere copiati nel file mqwebuser.xml .

## Backup e ripristino della configurazione del server mqweb

È possibile eseguire il backup della configurazione del server mqweb e ripristinarla nella stessa ubicazione o in un'ubicazione diversa.

### Prima di iniziare

Prima di poter ripristinare la configurazione del server mqweb, è necessario installare IBM MQ, o il IBM MQ Web Server autonomo, sul sistema in cui si desidera ripristinare il server mqweb. In un'installazione IBM MQ Web Server autonoma, è necessario creare il server mqweb attenendosi alla procedura descritta in [“Configurazione del IBM MQ Web Server autonomo”](#) a pagina 842.

### Informazioni su questa attività

Seguire la procedura in questa attività per eseguire il backup e ripristinare la configurazione del server mqweb. Se si ripristina il server mqWeb in un'ubicazione differente, è necessario aggiornare la configurazione del server mqWeb per assicurarsi che i riferimenti ai file siano corretti.

**V 9.4.0** È inoltre possibile utilizzare questa procedura per migrare un server mqWeb attualmente in esecuzione in un'installazione IBM MQ da eseguire in un'installazione IBM MQ Web Server autonoma.

### Procedura

- Per eseguire il backup della configurazione del server mqweb, copiare tutti i file nella directory che contiene la configurazione del server mqweb nella propria ubicazione di backup.
  - In un'installazione IBM MQ , copiare il contenuto della seguente directory:
    - **Linux** / **AIX** Su AIX o Linux: `/var/mqm/web/installations/installationName`
    - **Windows** Su Windows: `MQ_DATA_PATH\web\installations\installationName`, dove `MQ_DATA_PATH` è il percorso dati IBM MQ . Questo percorso è il percorso dati



selezionato durante l'installazione di IBM MQ. Per impostazione predefinita, questo è C:\ProgramData\IBM\MQ.

-  Su z/OS: la directory utente WebSphere Liberty specificata quando lo script **crtmqweb** è stato eseguito per la creazione della definizione del server mqweb.
  -   In un'installazione autonoma di IBM MQ Web Server, copiare il contenuto della directory `MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST`, dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente `MQ_OVERRIDE_DATA_PATH`.
2. Per ripristinare la configurazione del server mqweb, sostituire il contenuto della directory che contiene la configurazione del server mqweb con i file copiati al passo “1” a pagina 872.
- In un'installazione IBM MQ, sostituire il contenuto della directory seguente:
    -   Su AIX o Linux: `/var/mqm/web/installations/installationName`
    -  Su Windows: `MQ_DATA_PATH\web\installations\installationName`, dove `MQ_DATA_PATH` è il percorso dati IBM MQ. Questo percorso è il percorso dati selezionato durante l'installazione di IBM MQ. Per impostazione predefinita, questo è C:\ProgramData\IBM\MQ.
    -  Su z/OS: la directory utente WebSphere Liberty specificata quando lo script **crtmqweb** è stato eseguito per la creazione della definizione del server mqweb.
  -   In un'installazione autonoma di IBM MQ Web Server, sostituire il contenuto della directory `MQ_OVERRIDE_DATA_PATH/web/installations/MQWEBINST`, dove `MQ_OVERRIDE_DATA_PATH` è la directory di dati IBM MQ Web Server a cui fa riferimento la variabile di ambiente `MQ_OVERRIDE_DATA_PATH`.
3. Impostare la proprietà dei file ripristinati nel passo “2” a pagina 873 in modo tale che l'ID utente del server mqweb possa accedervi.
4. Se la configurazione del server mqweb è stata ripristinata in un'altra posizione, modificare il valore di tutte le proprietà nella configurazione del server mqweb che fanno riferimento ai file nella precedente directory di configurazione del server mqweb.
- a) Prima di immettere i comandi **setmqweb** o **dspmqweb**, impostare l'ambiente in modo che punti alla propria configurazione del server mqweb.
-  Su z/OS, impostare la variabile di ambiente **WLP\_USER\_DIR** in modo che la variabile 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 al comando **crtmqweb**.  
Per ulteriori informazioni, consultare [Creazione del server mqweb](#).
  -   In una installazione IBM MQ Web Server autonoma, impostare la variabile di ambiente `MQ_OVERRIDE_DATA_PATH` sulla directory di dati IBM MQ Web Server.
  - In tutti gli altri ambienti, non è necessario completare alcuna azione per impostare l'ambiente.
- b) Visualizzare il valore di tutte le proprietà configurabili del server mqweb che un utente ha modificato. Emetti il seguente comando:

```
dspmqweb properties -u
```
- c) Se viene visualizzata la proprietà **remoteKeyfile**, verificare il valore della proprietà.  
Se il valore della proprietà fa riferimento a un percorso file nella precedente directory di configurazione del server mqweb, modificare il valore per fare riferimento al percorso file

nella nuova directory di configurazione del server mqweb. Immettere il seguente comando per modificare il valore della proprietà **remoteKeyfile** :

```
setmqweb properties -k remoteKeyfile -v path_to_keyfile
```

d) Visualizzare la configurazione del gestore code remoto del server mqweb. Emetti il seguente comando:

```
dspmweb remote -a
```

e) Se viene visualizzata una delle seguenti proprietà, controllare il valore della proprietà:

- **globalTrustStorePath**
- **globalKeyStorePath**
- **ccdtURL**
- **keyStorePath**
- **trustStorePath**

Modificare il valore di qualsiasi proprietà che fa riferimento a un percorso file nella directory di configurazione del server mqweb precedente per fare riferimento al percorso file nella nuova directory di configurazione del server mqweb. Immettere il comando **setmqweb remote** per modificare il valore di ciascuna proprietà. Ad esempio, per modificare il valore della proprietà **keyStorePath** per il gestore code remoto con il nome univoco **remote-QM1**, immettere il seguente comando:

```
setmqweb remote -uniqueName remote-QM1 -keyStorePath new_keystore_path
```

Per ulteriori informazioni, consultare [setmqweb remote \(set mqweb server remote queue manager configuration\)](#).

## MQ Adv. Linux Windows MQ Adv. VUE MQ Adv. z/OS **Definizione di una connessione Aspera gateway su piattaforme Linux o Windows**

IBM Aspera faspio Gateway fornisce un tunnel TCP/IP veloce che può aumentare in modo significativo la velocità di trasmissione di rete per IBM MQ. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linux o Windows.

### Informazioni su questa attività

Aspera gateway può essere utilizzato per migliorare le prestazioni dei canali del gestore code. È particolarmente efficace se la rete ha una latenza elevata o tende a perdere pacchetti ed è generalmente utilizzata per velocizzare la connessione tra gestori code in diversi data center.

**Nota:** Per una rete veloce che non perde pacchetti, si verifica una diminuzione delle prestazioni quando si utilizza Aspera gateway, quindi è importante controllare le prestazioni della rete prima e dopo la definizione di una connessione Aspera gateway .

Definire un Aspera gateway ad ogni estremità della connessione di rete IP, quindi utilizzare TCP/IP per connettere i canali del gestore code a ciascun gateway. Un gestore code non deve essere in esecuzione sulla stessa macchina del Aspera gateway che utilizza e più gestori code possono utilizzare lo stesso gateway.

Per utilizzare Aspera gateway, è necessario disporre di una o più delle seguenti titolarità:

- **MQ Adv.** IBM MQ Advanced for Multiplatforms
- **MQ Appliance** IBM MQ Appliance
- **MQ Adv. VUE** IBM MQ Advanced for z/OS VUE

- **MQ Adv. z/OS** IBM MQ Advanced for z/OS

È possibile distribuire Aspera gateway su una qualsiasi delle seguenti piattaforme:

- Linux for x86-64
- Linux on Power Systems - Little Endian
- Linux for IBM Z
- Windows - per ulteriori informazioni sul supporto della piattaforma su Windows, consulta la [documentazione di IBM Aspera faspio Gateway](#).

L'utilizzo di Aspera gateway è limitato ai messaggi IBM MQ a meno che il gateway non sia autorizzato separatamente.

I gestori code che utilizzano Aspera gateway possono essere in esecuzione su qualsiasi piattaforma supportata. Per un elenco completo delle piattaforme supportate, consultare [Icane utilizzate nella documentazione del prodotto](#).

Per ogni gestore code che non si trova sulla stessa macchina del Aspera gateway che utilizza, verificare di disporre di una connessione di rete veloce tra il gestore code e Aspera gateway.

Si utilizza un file tom1 per creare una definizione di gateway che definisce le porte in entrata e in uscita utilizzate dal gateway. Un file tom1 di esempio viene fornito con Aspera gateway. La definizione del gateway in entrata definisce la connessione dal gestore code locale al gateway e dal gateway locale a quello remoto. La definizione gateway in entrata definisce la connessione dal gateway remoto al gateway locale e dal gateway locale al gestore code locale.

La seguente procedura fornisce una guida di base per l'attività e l'esecuzione. Per informazioni più dettagliate, consultare la [documentazione di IBM Aspera faspio Gateway](#).

## Procedura

1. Ottenere l'immagine di installazione Aspera gateway .





► **Multi** Per Multiplatforms, si scarica Aspera gateway da Passport Advantage. Il download è etichettato "IBM Aspera faspio Continuous Delivery Release for IBM MQ V9.4 Multiplatform Multilingual eAssembly". Viene distribuito come immagine Continuous Delivery (CD) solo a causa del ritmo di modifica in questa area, il che significa che sono necessari aggiornamenti alla frequenza delle release CD ed è possibile installarlo su qualsiasi sistema IBM MQ che abbia la titolarità IBM MQ Advanced for Multiplatforms o IBM MQ Appliance . Per scaricare questo eAssembly, andare a [Download IBM MQ 9.4](#) , quindi fare clic sulla scheda per la release richiesta. L' eAssembly contiene le immagini di installazione per tutte le piattaforme su cui è disponibile il gateway. eAssembly contiene anche un file `ibm-faspio-license.zip` , che contiene un file di licenza.

► **MQ Adv. VUE** ► **MQ Adv. z/OS** Se il sistema IBM MQ dispone della titolarità IBM MQ Advanced for z/OS VUE o IBM MQ Advanced for z/OS , si ottiene il Aspera gateway dal componente Connector Pack che fa parte dell'installazione SMP/E.



► **MQ Adv. VUE** ► **MQ Adv. z/OS** I file per IBM MQ Advanced for z/OS VUE e IBM MQ Advanced for z/OS sono i seguenti:

*Tabella 53. Nomi file e numeri di versione faspio per piattaforma e versione IBM MQ*

Piattaforma	Nome file	numero versione faspio
Linux for x86-64	► <b>V 9.4.0</b> ► <b>V 9.4.0</b> MOGV DEN . zip	1.3.4
Linux on Power Systems - Little Endian	► <b>V 9.4.0</b> ► <b>V 9.4.0</b> MOGV FEN . zip	1.3.4

Tabella 53. Nomi file e numeri di versione faspio per piattaforma e versione IBM MQ (Continua)		
Piattaforma	Nome file	numero versione faspio
Linux for IBM Z	  M0GVGEN.zip	1.3.4
Windows	  M0GVHEN.zip	1.3.4

**Nota:** Aspera gateway non può essere eseguito in modo nativo su z/OS.

  Oltre alle immagini di installazione, la directory fasp contiene M05QKEN.zip, che contiene un file di licenza.

- Copiare l'immagine di installazione Aspera gateway sulle due macchine che eseguiranno il gateway, quindi estrarre e installare il gateway.  
Utilizzare il file di licenza contenuto in `ibm-faspio-license.zip` (Multiplatforms) o `M05QKEN.zip` (z/OS). Per ulteriori informazioni, consultare la documentazione IBM Aspera faspio Gateway :

-  [Installazione su Linux](#)

-  [Installazione su Windows](#)

- Configurare e proteggere ciascun gateway.

Per ulteriori informazioni, consultare la documentazione IBM Aspera faspio Gateway :

- [Configurazione del file di configurazione Gateway](#)
- [Protezione del gateway](#)

- Ad ogni estremità della connessione di rete, modificare la definizione di canale per connettersi alla porta su cui è in ascolto il gateway locale.

- Avviare ciascun servizio gateway.

Per ulteriori informazioni, consultare la documentazione IBM Aspera faspio Gateway :

-  [Avvio su Linux](#)

-  [Avvio su Windows](#)

- Riavviare i canali.

I gestori code stanno ora comunicando attraverso una connessione Aspera gateway .


### Esempio

Questo esempio definisce una connessione Aspera gateway su due computer su cui è in esecuzione Linux. La configurazione è la seguente:

- L'indirizzo IP della macchina gateway locale è 9.20.193.107. L'indirizzo IP della macchina gateway remota è 9.20.192.115.
- Il gestore code locale è in esecuzione su una macchina con indirizzo IP 9.20.121.5. Il gestore code remoto è in esecuzione su una macchina con indirizzo IP 9.20.121.25. Entrambi i gestori code sono in ascolto sulla porta 1414.
- Il canale del gestore code sul gestore code locale viene modificato per connettersi al Aspera gateway locale utilizzando **conname** 9.20.193.107(1500). Il canale del gestore code sul gestore code remoto è stato modificato per connettersi al Aspera gateway remoto utilizzando **conname** 9.20.192.115(1500).
- Da IBM Aspera faspio Gateway 1.2, TLS è abilitata per impostazione predefinita. Se vuoi configurare TLS con il gateway, vedi [Protezione del gateway](#) nella documentazione di IBM Aspera faspio Gateway .

1. Definire una connessione Aspera gateway sulla macchina gateway locale:

- Installare Aspera gateway:

–  Su Linux, utilizzare i seguenti comandi:

```
rpm -ivh ibm-faspio-gateway-<version>.x86_64.rpm
```

- Modificare il file di gateway .toml nella directory creata dall'installazione:

Modificare il file per impostare le definizioni del gateway locale.

```
[[bridge]]
 name = "Outbound"
 [bridge.local]
 protocol = "tcp"
 host = "9.20.193.107"
 port = 1500
 tls_enabled = false

 [bridge.forward]
 protocol = "fasp"
 host = "9.20.192.115"
 port = 1600
 tls_enabled = false

[[bridge]]
 name = "Inbound"
 [bridge.local]
 protocol = "fasp"
 host = "9.20.193.107"
 port = 1600
 tls_enabled = false

 [bridge.forward]
 protocol = "tcp"
 host = "9.20.121.5"
 port = 1414
 tls_enabled = false
```

- Copiare il file aspera-license da ibm-faspio-license.zip (Multiplatforms) o M05QKEN.zip (z/OS) in /usr/local/etc/faspio/.

2. Ripetere il passo precedente per definire una connessione Aspera gateway sulla macchina gateway remota.


- Modificare il file gateway .toml nella directory creata dall'installazione. Modificare il file per impostare le definizioni del gateway remoto:


```
[[bridge]]
 name = "Outbound"
 [bridge.local]
 protocol = "tcp"
 host = "9.20.193.107"
 port = 1500
 tls_enabled = false

 [bridge.forward]
 protocol = "fasp"
 host = "9.20.192.115"
 port = 1600
 tls_enabled = false

[[bridge]]
 name = "Inbound"
 [bridge.local]
 protocol = "fasp"
 host = "9.20.193.107"
 port = 1600
 tls_enabled = false

 [bridge.forward]
 protocol = "tcp"
 host = "9.20.121.5"
 port = 1414
 tls_enabled = false
```

- Copiare il file `aspera-license` da `ibm-faspio-license.zip` (Multiplatforms) o `M05QKEN.zip` (z/OS) in `/usr/local/etc/faspio/`.
3. Ad ogni estremità della connessione, modificare la definizione di canale per connettersi alla porta su cui è in ascolto il gateway locale.
- Modificare il canale del gestore code sul gestore code locale per connettersi al Aspera gateway locale utilizzando **conname** `9.20.193.107(1500)`.
  - Modificare il canale del gestore code sul gestore code remoto per connettersi al Aspera gateway remoto utilizzando **conname** `9.20.192.115(1500)`.
4. Avviare il gateway locale immettendo il seguente comando sulla macchina del gateway locale:
- 

```
sudo systemctl start faspio-gateway
```
5. Avviare il gateway remoto eseguendo il seguente comando sulla macchina del gateway remoto:
- 

```
sudo systemctl start faspio-gateway
```
6. Riavviare i canali.

## Operazioni successive

Aspera gateway trasmette i dati che riceve, senza interpretarli in alcun modo. Ciò significa che è possibile configurare TLS tra i canali del gestore code che utilizzano Aspera gateway perché la connessione gateway non è a conoscenza dell'handshake TLS. Ciò significa anche che i gestori code su qualsiasi piattaforma IBM MQ supportata possono utilizzare Aspera gateway.

Per utilizzare un gestore code a più istanze con un gateway, configurare le definizioni gateway per ogni istanza del gestore code.

**Nota:** Aspera gateway è stato verificato solo con i canali del gestore code. Non è stato testato con i canali client. Questo perché l'uso previsto per Aspera gateway è quello di connettere i gestori code remoti su una rete lenta, mentre le applicazioni client generalmente si connettono ai gestori code in un data center locale su una rete veloce.

### Riferimenti correlati

[“Il tipo di comunicazione da utilizzare” a pagina 17](#)

Piattaforme differenti supportano protocolli di comunicazione differenti. La scelta del protocollo di trasmissione dipende dalla combinazione di IBM MQ MQI client e delle piattaforme server.

[IBM Aspera faspio Gateway documentazione](#)

## Configurazione di IBM MQ per l'utilizzo con il servizio di misurazione IBM Cloud Private

Configurazione di IBM MQ per l'utilizzo con il servizio di misurazione IBM Cloud Private per il report e la visualizzazione delle informazioni di avvio e utilizzo del gestore code.

### Prima di iniziare

Prima di configurare i tuoi gestori code IBM MQ per utilizzare un servizio IBM Cloud Private, devi avere un account IBM Cloud. Per creare il tuo account, consulta [Registrati per IBM Cloud](#).

## Informazioni su questa attività

Utilizzando il [servizio di misurazione IBM Cloud Private](#), puoi connettere i tuoi prodotti IBM in loco alla tua istanza del servizio in IBM Cloud Private e visualizzare tutti i prodotti registrati nella tua organizzazione in un singolo dashboard.

È possibile configurare e collegare i gestori code AIX, Linux e Windows alla propria istanza del servizio di misurazione e visualizzare le informazioni di avvio e utilizzo. Tuttavia, su piattaforme diverse dagli ambienti Linux Container, i dati non possono essere utilizzati a supporto delle licenze di prezzo basate sui contenitori orari.

Per registrare i dati di utilizzo per un tipo di licenza VPC mensile, invece della metrica di licenza oraria predefinita, imposta la variabile di ambiente `AMQ_LICENSEING_METRIC=VPCMonthlyPeak`. Ciò fa sì che il gestore code carichi i dati relativi ai tipi di licenze VPC mensili, invece del comportamento predefinito di caricamento dei dati relativi alle licenze orarie basate sul contenitore.

Utilizzare i seguenti attributi con la stanza `ReportingService` nel file `qm.ini` :

### APIKeyFile

Ubicazione del file di testo con il valore dell'istanza del servizio di misurazione **APIKey** .

### CapacityReporting

Scrive periodicamente i messaggi di log di errori nei log AMQERR nel seguente formato:

```
4/22/2020 01:44:29 PM - Process(1274.1) User(bld-adm) Program(amqmgr0)
Host(8b3b83f2bc7d) Installation(Docker)
VRMF(9.2.0.0)
Time(2020-04-22T13:44:29.295Z)
ArithInsert1(300)
CommentInsert1(8.5)
CommentInsert2(IBM MQ Advanced)
```

Le informazioni prodotte dall'attributo **CapacityReporting** vengono inserite in un messaggio AMQ5064, che fornisce una migliore comprensione di quanto IBM MQ l'azienda sta utilizzando:

### AMQ5064

Questo gestore code è stato in esecuzione per 300 secondi. Attualmente è in esecuzione con 8.5 core. Il tipo di licenza è IBM MQ Advanced.

### Gravità

0: Informazioni

### Spiegazione

Questo è un messaggio informativo per la traccia dell'utilizzo.

### Risposta

Nessuna.

### LicensingGroup

Il gruppo di fatturazione a cui appartiene il gestore code. Ciò influenza il modo in cui i dati vengono raggruppati nei report generati dal servizio di misurazione.

### ServiceURL

L'indirizzo di servizio IBM Cloud Private .

### ServiceProxy

L'URL e la porta per il proxy HTTP che possono essere utilizzati se i gestori code non hanno accesso diretto alla rete su cui è in esecuzione il servizio di misurazione.

È possibile visualizzare gli host su cui sono installati i propri prodotti, le versioni del prodotto utilizzate e le piattaforme su cui sono in esecuzione. Dalle metriche di utilizzo di alto livello che vengono visualizzate per ogni prodotto, puoi avere una panoramica di quanto sono pesanti i carichi di lavoro. Per IBM MQ, puoi vedere quali gestori code sono più utilizzati e quali hanno carichi di lavoro più leggeri.

Quando un gestore code è configurato per connettersi a un'istanza del servizio di misurazione, le seguenti informazioni sono riportate a IBM Cloud Private:

- IBM MQ Nome gestore code

- IBM MQ Identificativo gestore code
- Directory root di installazione IBM MQ
- Componenti installati IBM MQ (nome e versione)
- Nome host
- Nome sistema operativo host
- Versione sistema operativo host
- Informazioni sull'utilizzo del VPC (Virtual Processor Core) per il gestore code IBM MQ

Puoi monitorare le metriche di utilizzo del tuo VPC del gestore code nel tuo pannello di controllo dell'istanza del servizio di misurazione.

## Procedura

- Configurare un gestore code da utilizzare con l'istanza del servizio di misurazione su IBM Cloud Private.
- Connettersi al servizio di misurazione IBM Cloud Private tramite proxy HTTP.
- Risolvere i problemi di connessione al servizio di misurazione di IBM Cloud Private .

### Riferimenti correlati

[Metrica dei prezzi per VPC \(Virtual Processor Core\)](#)

## **Multi** Configurazione di un gestore code per l'utilizzo con l'istanza del servizio di misurazione su IBM Cloud Private

Imposta le informazioni di sicurezza e di registrazione IBM Cloud per il tuo gestore code e connettiti all'istanza del servizio di misurazione che hai già creato.

### Informazioni su questa attività

Il dashboard dell'istanza del servizio di misurazione IBM Cloud Private mostra i dati solo per i gestori code configurati per includere la sicurezza e le informazioni di registrazione IBM Cloud Private .

## Procedura

1. Seguire la procedura documentata IBM Cloud Private per creare un ID servizio all'indirizzo: [Creazione di un ID servizio utilizzando la CLI IBM Cloud Private](#).
2. Atteniti alla procedura documentata IBM Cloud Private per creare una chiave API all'indirizzo: [API di gestione delle chiavi API](#).
3. Scarica i certificati TLS dal cluster IBM Cloud Private .  
Prendi nota dell'ubicazione in cui hai scaricato i certificati. È possibile aggiungere i certificati scaricati al repository delle chiavi per il gestore code, nel passo “9” a pagina 881.
4. Creare un file di testo `apikeyfile.txt` e aggiungere il valore **API key** copiato nell'attività precedente.  
Nota l'ubicazione di `apikeyfile.txt` in modo da poter includere il relativo percorso nel passo 8. Questo file deve essere leggibile dall'utente del gestore code (`'mq'` su sistemi AIX and Linux ). Il file deve contenere solo il **API key** stesso, non un payload JSON, ad esempio `d9c11b45-4dda-4de4-c0b2-2e4e1004dc64`
5. Creare il gestore code, ad esempio `QM1`.  
Per ulteriori informazioni, fare riferimento a [Creazione e gestione di gestori code su più piattaforme](#).
6. Avviare il gestore code `QM1`.  
Per ulteriori informazioni, consultare [Avvio di un gestore code](#).



7. Ricordati di impostare il tuo ambiente della riga di comando IBM MQ prima di eseguire comandi IBM MQ.

Eseguire il comando **setmqenv**.

**AIX** Su AIX:

```
. /usr/mqm/bin/setmqenv -s
```

**Linux** Su Linux:

```
. /opt/mqm/bin/setmqenv -s
```

**Windows** Su Windows:

```
"C:\Program Files\IBM\MQ\bin\setmqenv.cmd" -n installation name
```

8. Creare un truststore SSL per il gestore code QM1.

**AIX** Iniziare la creazione del truststore, su AIX:

```
runmqakm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-expire 30 -stash
```

**Linux** Su Linux:

```
runmqakm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-expire 30 -stash
```

**Windows** Su Windows:

```
runmqakm -keydb -create -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type
cms -expire 30 -stash
```

9. Aggiungere i certificati digitali scaricati nel passo [“3”](#) a pagina 880 al repository delle chiavi del gestore code.

**AIX** Su AIX:

```
runmqakm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label RootCA
-file Download_location/RootCA.crt -format ascii -trust enable
```

```
runmqakm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label ServerCert
-file Download_location/CERT.crt -format ascii -trust enable
```

**Linux** Su Linux:

```
runmqakm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label RootCA
-file Download_location/RootCA.crt -format ascii -trust enable
```

```
runmqakm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms
-label ServerCert
-file Download_location/CERT.crt -format ascii -trust enable
```

**Windows** Su Windows:

```
runmqakm -cert -add -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms
-label RootCA
-file "Download_location\RootCA.crt" -format ascii -trust enable
```

```
runmqakm -cert -add -db "C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key.kdb" -pw password -type
cms -label ServerCert
-file "Download_location\CERT.crt" -format ascii -trust enable
```

10. Aggiungere la nuova stanza `ReportingService` con il percorso `apikeyfile` al file `qm.ini` del gestore code:

```
ReportingService:
 APIKeyFile=APIKey file location/apapikeyfile.txt
```

11. Aggiungere il valore **API host** al file `qm.ini`.

La stanza `ReportingService` contiene ora il percorso ai valori `apikeyfile` e **API host** (**ServiceURL**):

```
ReportingService:
 APIKeyFile=APIKey file location/apapikeyfile.txt
 ServiceURL=https://productinsights-api.ng.bluemix.net
```

Salvare e uscire da `qm.ini`.

12. Riavviare il gestore code per rendere effettive le modifiche.  
Potrebbe essere richiesto di concedere l'autorizzazione per il processo del gestore code `amqzmu0` per accedere alla rete. L'accesso è richiesto per abilitare il gestore code a contattare il servizio di misurazione.
13. Visualizzare informazioni sul gestore code `QM1` nell'istanza del servizio di misurazione.  
Quando lo stato di creazione report è attivo, le informazioni di avvio e utilizzo per tutti i server di integrazione sul nodo di integrazione specificato vengono riportate al servizio di misurazione. Le informazioni di utilizzo vengono aggiornate ogni 15 minuti.
14. Opzionale: Arrestare un gestore code dalla notifica al servizio di misurazione, rimuovendo la sezione `ReportingService` dal file `qm.ini` del gestore code e riavviare il gestore code.
15. Opzionale: Controllare le informazioni diagnostiche nel file di log del gestore code se il gestore code non riesce a riportare le informazioni di avvio o di utilizzo al servizio di misurazione.

**AIX**

Su AIX:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

**Linux**

Su Linux:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

**Windows**

Su Windows:

```
C:\ProgramData\IBM\MQ\errors\AMQERR0*.log
```

## Risultati

Hai creato un'istanza del servizio di misurazione e configurato il gestore code per la connessione all'istanza. È possibile visualizzare le informazioni sul gestore code nel dashboard dell'istanza del servizio di misurazione.

**Multi**

## Connessione al servizio di misurazione IBM Cloud Private tramite un proxy HTTP

Se il tuo gestore code è in esecuzione su un sistema che non ha accesso diretto al tuo cluster ICP, puoi utilizzare un proxy HTTP fornito dalla tua organizzazione per connetterti alla tua istanza del servizio di misurazione in IBM Cloud Private.

### Prima di iniziare

È stata configurata la sicurezza, è stato aggiunto l'URL del servizio e **API key** al file `qm.ini` del proprio gestore code.

## Informazioni su questa attività

Utilizza questa attività per configurare il tuo gestore code per la connessione all'istanza del [servizio di misurazione](#) in IBM Cloud Private tramite un proxy HTTP fornito dalla tua organizzazione.

### Procedura

- Aggiungere un attributo proxy del servizio alla stanza di registrazione IBM Cloud Private del proprio file `qm.ini`.

È possibile impostare l'attributo **ServiceProxy** come segue:

- Un URL che include il prefisso `http://` e, facoltativamente, la porta. Se non si specifica la porta, viene utilizzato `1080`.

```
ReportingService:
ServiceProxy=http://myorgproxy.net:1080
```

**Nota:** Il parametro **ServiceProxy** deve essere impostato su un URL `http://` valido. Altri protocolli proxy, ad esempio, HTTPS e SOCKS non sono supportati.

- Riavviare il gestore code prima che le modifiche diventino effettive.

Multi

## Risoluzione dei problemi di connessione al servizio di misurazione

Avviso di risoluzione dei problemi per gli errori che potresti riscontrare durante la connessione del tuo gestore code a un'istanza del servizio di misurazione.

### Il gestore code non può registrare o caricare le metriche di utilizzo nel servizio di misurazione configurato

Verificare che il gestore code abbia accesso alla rete. Il valore **APIKey** nel file della chiave API non è corretto. Assicurarsi che il componente IBM Global Security Kit (GSKit) sia installato.

### Stanza `qm.ini` non valida

È stata trovata una stanza `qm.ini` non valida. Per ulteriori informazioni, consultare il log degli errori.

### Parametro proxy servizio HTTP non valido

Il valore per l'attributo **ServiceProxy** per la sezione del gestore code `ReportingService` non è configurato correttamente. Il gestore code non si registra con il servizio. Il parametro **ServiceProxy** deve essere impostato su un URL `http://` valido. Altri protocolli proxy, ad esempio, HTTPS e SOCKS non sono supportati.

z/OS

## Configuring queue managers on z/OS

Use these instructions to configure queue managers on IBM MQ for z/OS.

### Before you begin

Before you configure IBM MQ for z/OS, read:

- [IBM MQ for z/OS concepts](#)
- [Planning your IBM MQ environment on z/OS](#)

### About this task

After you have installed IBM MQ, you must carry out a number of tasks before you can make it available to users.

## Procedure

- See the following subtopics for information on how to configure queue managers on IBM MQ for z/OS.

### Related concepts

 [Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS](#)

### Related tasks

[“Creazione di gestori code su più piattaforme” on page 7](#)

Prima di poter utilizzare messaggi e code, è necessario creare e avviare almeno un gestore code e i relativi oggetti associati. Un gestore code gestisce le risorse associate ad esso, in particolare le code di sua proprietà. Fornisce servizi di accodamento alle applicazioni per chiamate e comandi MQI (Message queuing Interface) per creare, modificare, visualizzare ed eliminare oggetti IBM MQ .

#### Securing

[“Configurazione dell'accodamento distribuito” on page 205](#)

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione


[“Configurazione delle connessioni tra client e server” on page 16](#)

Per configurare i link di comunicazione tra IBM MQ MQI clients e server, decidere il protocollo di comunicazione, definire le connessioni ad entrambe le estremità del link, avviare un listener e definire canali.

 [Administering IBM MQ for z/OS](#)

#### Planning

### Related reference

 [Using the IBM MQ for z/OS utilities](#)

## Preparing to customize queue managers on z/OS

Use this topic when customizing your queue managers with details of installable features, national language features, and information about testing, and setting up security.

### Preparing for customization

The Program Directory lists the contents of the IBM MQ installation tape, the program and service level information for IBM MQ, and describes how to install IBM MQ for z/OS using System Modification Program Extended (SMP/E). Per i collegamenti di download per le directory del programma, consultare [IBM MQ for z/OS Program Directory PDF files](#).

When you have installed IBM MQ, you must carry out a number of tasks before you can make it available to users. See the following sections for a description of these tasks:

- [“Setting up IBM MQ for z/OS” on page 888](#)
- [“Testing a queue manager on z/OS” on page 953](#)
- [Setting up security on z/OS](#)

If you are migrating from a previous version of IBM MQ for z/OS, you do not need to perform most of the customization tasks. See [Maintaining and migrating](#) for more information about the tasks you must perform.

### Installable features of IBM MQ for z/OS

IBM MQ for z/OS comprises the following features:

#### Base

This is required; it comprises all the main functions, including

- Administration and utilities

- Support for CICS, IMS, and batch type applications using the IBM MQ Application Programming Interface, or C++
- Distributed queuing facility (supporting both TCP/IP and APPC communications)

### National language features

These contain error messages and panels in all the supported national languages. Each language has a language letter associated with it. The languages and letters are:

- C**  
Simplified Chinese
- E**  
U.S. English (mixed case)
- F**  
French
- K**  
Japanese
- U**  
U.S. English (uppercase)

You must install the US English (mixed case) option. You can also install one or more other languages. (The installation process for other languages requires US English (mixed case) to be installed, even if you are not going to use US English (mixed case).)

### IBM MQ for z/OS UNIX System Services Components

This feature is optional. Select this feature if you want to build and run Java applications that use [Jakarta Messaging 3.0](#) or Java Message Service 2.0 to connect to IBM MQ for z/OS.

See [IBM MQ for z/OS Program Directory PDF files](#) for information on installing IBM MQ for z/OS UNIX System Services Components.

### IBM MQ for z/OS UNIX System Services Web Components

This feature is optional.

Select this feature if you want to use the IBM MQ Console, or the REST API.

You must install the IBM MQ for z/OS UNIX System Services Components feature, to install this feature.

### IBM MQ for z/OS Managed File Transfer

This feature is optional, and should only be installed if you have entitlement for IBM MQ Advanced for z/OS, IBM MQ for z/OS Value Unit Edition (VUE), or IBM MQ for z/OS Managed File Transfer.

Select this feature if you want to use the Managed File Transfer capabilities of IBM MQ for z/OS.

You must install the IBM MQ for z/OS UNIX System Services Components feature, to install this feature.

### Libraries that exist after installation

IBM MQ is supplied with a number of separate load libraries. [Table 54 on page 885](#) shows the libraries that might exist after you have installed IBM MQ.

<i>Table 54. IBM MQ libraries that exist after installation</i>	
<b>Name</b>	<b>Description</b>
thlqual.SCSQANLC	Contains the load modules for the Simplified Chinese version of IBM MQ.
thlqual.SCSQANLE	Contains the load modules for the U.S. English (mixed case) version of IBM MQ.
thlqual.SCSQANLF	Contains the load modules for the French version of IBM MQ.

Table 54. IBM MQ libraries that exist after installation (continued)

Name	Description
thlqual.SCSQANLK	Contains the load modules for the Japanese version of IBM MQ.
thlqual.SCSQANLU	Contains the load modules for the U.S. English (uppercase) version of IBM MQ.
thlqual.SCSQASMS	Contains source for assembler sample programs.
thlqual.SCSQAUTH	The main repository for all IBM MQ product load modules; it also contains the default parameter module, CSQZPARM. This library must be APF-authorized and in PDS-E format.
thlqual.SCSQCICS	Contains extra load modules that must be included in the CICS DFHRPL concatenation. This library must be APF-authorized and in PDS-E format.
thlqual.SCSQCLST	Contains CLISTS used by the sample programs.
thlqual.SCSQCOBC	Contains COBOL copybooks, including copybooks required for the sample programs.
thlqual.SCSQCOBS	Contains source for COBOL sample programs.
thlqual.SCSQCPPS	Contains source for C++ sample programs.
thlqual.SCSQC37S	Contains source for C sample programs.
thlqual.SCSQC370	Contains C headers, including headers required for the sample programs.
thlqual.SCSQDEFS	Contains side definitions for C++ and the Db2 DBRMs for shared queuing.
thlqual.SCSQEXEC	Contains REXX executable files to be included in the SYSEXEC or SYSPROC concatenation if you are using the IBM MQ operations and control panels.
thlqual.SCSQFCMD	Contains templates for jobs to create and run Managed File Transfer tasks.
thlqual.SCSQHPPS	Contains header files for C++.
thlqual.SCSQINST	Contains JCL for installation jobs.
thlqual.SCSQLINK	Early code library. Contains the load modules that are loaded at system initial program load (IPL). The library must be APF-authorized.
thlqual.SCSQLOAD	Load library. Contains load modules for non-APF code, user exits, utilities, samples, installation verification programs, and adapter stubs. The library does not need to be APF-authorized and does not need to be in the link list. This library must be in PDS-E format.
thlqual.SCSQMACS	Contains Assembler macros including: sample macros, product macros, and system parameter macros.
thlqual.SCSQMAPS	Contains CICS mapsets used by sample programs.
thlqual.SCSQMSGC	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the Simplified Chinese language feature for the IBM MQ operations and control panels.

Table 54. IBM MQ libraries that exist after installation (continued)

Name	Description
thlqual.SCSQMSGE	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the U.S. English (mixed case) language feature for the IBM MQ operations and control panels.
thlqual.SCSQMSGF	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the French language feature for the IBM MQ operations and control panels.
thlqual.SCSQMSGK	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the Japanese language feature for the IBM MQ operations and control panels.
thlqual.SCSQMSGU	Contains ISPF messages to be included in the ISPMLIB concatenation if you are using the U.S. English (uppercase) language feature for the IBM MQ operations and control panels.
thlqual.SCSQMVR1	Contains the load modules for distributed queuing. This library must be APF-authorized and in PDS-E format.
thlqual.SCSQPLIC	Contains PL/I include files.
thlqual.SCSQPLIS	Contains source for PL/I sample programs.
thlqual.SCSQPnLA	Contains IPCS panels, for the dump formatter, to be included in the ISPPLIB concatenation. Also contains panels for IBM MQ sample programs.
thlqual.SCSQPnLC	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the Simplified Chinese language feature for the IBM MQ operations and control panels.
thlqual.SCSQPnLE	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the U.S. English (mixed case) language feature for the IBM MQ operations and control panels.
thlqual.SCSQPnLF	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the French language feature for the IBM MQ operations and control panels.
thlqual.SCSQPnLK	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the Japanese language feature for the IBM MQ operations and control panels.
thlqual.SCSQPnLU	Contains ISPF panels to be included in the ISPPLIB concatenation if you are using the U.S. English (uppercase) language feature for the IBM MQ operations and control panels.
thlqual.SCSQPROC	Contains sample JCL and default system initialization data sets.
thlqual.SCSQSNLC	Contains the load modules for the Simplified Chinese versions of the IBM MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQSNLE	Contains the load modules for the U.S. English (mixed case) versions of the IBM MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQSNLF	Contains the load modules for the French versions of the IBM MQ modules that are required for special purpose function (for example the early code).

Name	Description
thlqual.SCSQSNLK	Contains the load modules for the Japanese versions of the IBM MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQSNLU	Contains the load modules for the U.S. English (uppercase) versions of the IBM MQ modules that are required for special purpose function (for example the early code).
thlqual.SCSQTBLC	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the Simplified Chinese language feature for the IBM MQ operations and control panels.
thlqual.SCSQTBLE	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the U.S. English (mixed case) language feature for the IBM MQ operations and control panels.
thlqual.SCSQTBLF	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the French language feature for the IBM MQ operations and control panels.
thlqual.SCSQTBLK	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the Japanese language feature for the IBM MQ operations and control panels.
thlqual.SCSQTBLU	Contains ISPF tables to be included in the ISPTLIB concatenation if you are using the U.S. English (uppercase) language feature for the IBM MQ operations and control panels.

**Note:** Do not modify or customize any of these libraries. If you want to make changes, copy the libraries and make your changes to the copies.

### Related concepts

[IBM MQ for z/OS concepts](#)

[“Using IBM MQ with IMS” on page 991](#)

The IBM MQ -IMS adapter, and the IBM MQ - IMS bridge are the two components which allow IBM MQ to interact with IMS.

[“Using IBM MQ with CICS” on page 999](#)

To use IBM MQ with CICS, you must configure the IBM MQ CICS adapter and, optionally, the IBM MQ CICS bridge components.

[“Using OTMA exits in IMS” on page 1001](#)

Use this topic if you want to use IMS Open Transaction Manager Access exits with IBM MQ for z/OS.

### Related tasks

[“Setting up communications with other queue managers on z/OS” on page 961](#)

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

[Administering IBM MQ for z/OS](#)

### Related reference

[“Upgrading and applying service to Language Environment or z/OS Callable Services” on page 999](#)

The actions you must take vary according to whether you use CALLLIBS or LINK, and your version of SMP/E.

## Setting up IBM MQ for z/OS

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .



The best way to configure a queue manager is to carry out the following steps in the order shown:

1. Configure the base queue manager.
2. Configure the channel initiator, which performs queue manager to queue manager communications, and remote client application communication.
3. If you want to encrypt or protect messages, configure Advanced Message Security for z/OS.
4. If you want to use IBM MQ to transfer files, configure Managed File Transfer for z/OS.
5. If you want to use the administrative or messaging REST API, or the IBM MQ Console to manage IBM MQ from a web browser, configure the mqweb server.

This topic leads you through the various stages of setting up IBM MQ after you have successfully installed it. The installation process is described in the Program Directory. Per i collegamenti di download per le directory del programma, consultare [IBM MQ for z/OS Program Directory PDF files](#).

Samples are supplied with IBM MQ to help you with your customization. The sample data set members have names beginning with the four characters CSQ4 and are in the library thlqual.SCSQPROC.

Before you perform the customization tasks described in this topic, there are a number of configuration options that you must consider because they affect the performance and resource requirements of IBM MQ for z/OS. For example, you must decide which globalization libraries you want to use.

If you want to automate some of the customization steps, see [“Using IBM z/OSMF to automate IBM MQ” on page 1005](#).

## Configuration options

For more information about these options, see [Pianificazione su z/OS](#).

The description of each task in this section indicates whether:

- The task is part of the process of setting up IBM MQ. That is, you perform the task once when you customize IBM MQ on the z/OS system. (In a parallel sysplex, you must perform the task for each z/OS system in the sysplex, and ensure that each z/OS system is set up identically.)
- The task is part of adding a queue manager. That is, you perform the task once for each queue manager when you add that queue manager.

None of the tasks require you to perform an IPL of your z/OS system, if you use commands to change the various z/OS system parameters, and perform [“Update SYS1.PARMLIB members” on page 903](#) as suggested.

To simplify operations and to aid with problem determination, ensure that all z/OS systems in a sysplex are set up identically, so that queue managers can be quickly created on any system in an emergency.

For ease of maintenance, consider defining aliases to refer to your IBM MQ libraries; for more information, see [Using an alias to refer to an IBM MQ library](#).

### Related concepts

[IBM MQ for z/OS concepts](#)

[“Using IBM MQ with IMS” on page 991](#)

The IBM MQ -IMS adapter, and the IBM MQ - IMS bridge are the two components which allow IBM MQ to interact with IMS.

[“Using IBM MQ with CICS” on page 999](#)

To use IBM MQ with CICS, you must configure the IBM MQ CICS adapter and, optionally, the IBM MQ CICS bridge components.

[“Using OTMA exits in IMS” on page 1001](#)

Use this topic if you want to use IMS Open Transaction Manager Access exits with IBM MQ for z/OS.

### Related tasks

[“Setting up communications with other queue managers on z/OS” on page 961](#)

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

### Related reference

[“Upgrading and applying service to Language Environment or z/OS Callable Services” on page 999](#)  
The actions you must take vary according to whether you use CALLLIBS or LINK, and your version of SMP/E.

## **Configuring the z/OS system for IBM MQ**

Use these topics as a step by step guide for customizing your IBM MQ for z/OS system.

### **Identify the z/OS system parameters**

Some of the tasks involve updating the z/OS system parameters. You need to know which ones were specified when the system IPL was performed.

- *You need to perform this task once for each z/OS system where you want to run IBM MQ.*
- *You might need to perform this task when migrating from a previous version.*

SYS1.PARMLIB(IEASYSpp) contains a list of parameters that point to other members of SYS1.PARMLIB (where pp represents the z/OS system parameter list that was used to perform an IPL of the system).

The entries you need to find are:

#### **For “L'APF autorizza le librerie di caricamento IBM MQ” on page 890:**

PROG=xx or APF=aa point to the Authorized Program Facility (APF) authorized library list (member PROGxx or IEFAPFaa)

#### **For “Aggiornare l'elenco di link z/OS e LPA” on page 891:**

LNK=kk points to the link list (member LNKLSTkk) LPA=mm points to the LPA list (member LPALSTmm)

#### **For “Update the z/OS program properties table” on page 895:**

SCH=xx points to the Program Properties Table (PPT) (member SCHEDxx)

#### **For “Define the IBM MQ subsystem to z/OS” on page 896:**

SSN=ss points to the defined subsystem list (member IEFSSNs)

## **L'APF autorizza le librerie di caricamento IBM MQ**

Autorizzazione APF per varie librerie. Alcuni moduli di caricamento potrebbero già essere autorizzati.

### Note:

- È necessario eseguire questa attività una volta per ciascuno z/OS sistema in cui desideri eseguire IBM MQ .
- Se utilizzi gruppi di condivisione code, devi assicurarti che le impostazioni per IBM MQ sono identici su ciascuno z/OS sistema nel sysplex.
- Potrebbe essere necessario eseguire questa attività durante la migrazione da una versione precedente.
- Utilizzo della Biblioteca Look Aside (LLA):
  - Alcuni IBM MQ l'utilizzo può causare un elevato input/output (IO) per caricare i moduli dalle librerie. Questo IO può essere ridotto utilizzando la funzione LLA del sistema operativo.
  - Questo IO elevato può verificarsi durante:
    - Applicazioni con una velocità MQCONN/MQDISC elevata, ad esempio in una procedura memorizzata WLM.
    - Caricamento delle uscite del canale. Se disponi di canali che si avviano e si interrompono frequentemente e utilizzano le uscite del canale.
  - Il membro CSVLLAxx in SYS1.PARMLIB specifica l'impostazione LLA. L'inclusione del nome di una libreria nell'istruzione LIBRARIES significa che una copia del programma verrà sempre presa da VLF (Virtual Lookaside Facility) e quindi solitamente non richiederà I/O se utilizzata in modo intensivo.

L'inclusione nell'istruzione FREEZE significa che non ci sono I/O per ottenere le directory di concatenazione delle istruzioni DD rilevanti (questo può spesso essere più I/O del caricamento del programma stesso).

Utilizzare il comando del sistema operativo " F LLA, REFRESH" dopo qualsiasi modifica a una di queste librerie.

Le IBM MQ librerie di caricamento thlqual.SCSQAUTH e thlqual.SCSQLINK devono essere autorizzate da APF. È inoltre necessario autorizzare APF le librerie per la funzione lingua nazionale (thlqual.SCSQANLx e thlqual.SCSQSNLx) e per la funzione di accodamento distribuito (thlqual.SCSQMVR1).

Tuttavia, tutti i moduli di caricamento in LPA sono automaticamente autorizzati APF. Così sono tutti i membri dell'elenco di link se SYS1.PARMLIB membro IEASYSpp contiene l'istruzione:

```
LNKAUTH=LNKLST
```

LNKAUTH=LNKLST è il valore predefinito se LNKAUTH non è specificato.

A seconda di ciò che si sceglie di inserire nell'LPA o nell'elenco di collegamenti (consultare [“Aggiornare l'elenco di link z/OS e LPA” a pagina 891](#) ), potrebbe non essere necessario inserire le librerie nell'elenco di collegamenti APF

**Nota:** È necessario autorizzare con APF tutte le librerie incluse in IBM MQ STEPLIB. Se si inserisce una libreria non autorizzata ad APF in STEPLIB, l'intera concatenazione di librerie perde la relativa autorizzazione APF.

Gli elenchi APF si trovano in SYS1.PARMLIB membro PROGxx o IEAAPFaa. Gli elenchi contengono i nomi delle librerie z/OS autorizzate APF. L'ordine delle voci negli elenchi non è significativo. Consultare [Elenco librerie autorizzate APF](#) per informazioni sugli elenchi APF.

Per ulteriori informazioni sull'ottimizzazione del sistema, consultare [SupportPac MP16](#)

Se si utilizzano membri PROGxx con formato dinamico, è necessario immettere solo il comando z/OS SETPROG APF, ADD, DSNAME=h1q.SCSQ XXXX, VOLUME= YYYYYY per rendere effettive le modifiche: dove XXXX varia in base al nome della libreria e dove YYYYYY è il volume. Altrimenti, se si utilizza il formato statico o i membri IEAAPFaa, è necessario eseguire un IPL sul sistema.

Tenere presente che è necessario utilizzare il nome effettivo della libreria nell'elenco APF. Se si tenta di utilizzare l'alias del dataset della libreria, l'autorizzazione ha esito negativo.

### Concetti correlati

[“Aggiornare l'elenco di link z/OS e LPA” a pagina 891](#)

Aggiornare le librerie LPA con la nuova versione delle prime librerie di codice. L'altro codice può essere inserito nell'elenco dei link o nell'area LPA.

[“Preparing to customize queue managers on z/OS” a pagina 884](#)

Use this topic when customizing your queue managers with details of installable features, national language features, and information about testing, and setting up security.

## **Aggiornare l'elenco di link z/OS e LPA**

Aggiornare le librerie LPA con la nuova versione delle prime librerie di codice. L'altro codice può essere inserito nell'elenco dei link o nell'area LPA.

- È necessario eseguire questa attività una volta per ogni sistema z/OS in cui si desidera eseguire IBM MQ.
- Se si utilizzano gruppi di condivisione code, è necessario aggiornare il codice iniziale in ciascun gestore code in QSG al livello IBM MQ 9.4.0 prima di migrare uno qualsiasi dei gestori code a IBM MQ 9.4.0.

Installare l'ultimo codice iniziale su ogni LPAR, quindi aggiornare i gestori code uno alla volta prima della migrazione. Non è necessario migrare tutti i gestori code contemporaneamente.

- Potrebbe essere necessario eseguire questa attività quando si esegue la migrazione da una versione precedente. Per ulteriori dettagli, consultare Program Directory. Per i collegamenti di download per le directory del programma, consultare [IBM MQ for z/OS Program Directory PDF files](#).

**Nota:** Il dataset per LPA è specifico della versione. Se si sta utilizzando un LPA esistente nel sistema, contattare l'amministratore di sistema per decidere quale LPA utilizzare.

## Codice iniziale

Alcuni moduli di caricamento IBM MQ devono essere aggiunti a MVS perché IBM MQ agisca come un sottosistema. Questi moduli sono noti come codice Early e possono essere eseguiti anche se un gestore code non è attivo. Ad esempio, quando un comando operatore viene emesso sulla console con un prefisso di comando IBM MQ , questo codice Early acquisirà il controllo e controllerà se è necessario avviare un gestore code o passare la richiesta a un gestore code in esecuzione. Questo codice viene caricato in LPA (Link Pack Area). Esiste una serie di moduli Early, utilizzati per tutti i gestori code, che devono essere al livello più alto di IBM MQ. Il codice iniziale proveniente da una versione superiore di IBM MQ funzionerà con un gestore code con una versione inferiore di IBM MQ, ma non il contrario.

Il codice iniziale comprende i seguenti moduli di caricamento:

- CSQ3INI e CSQ3EPX nella libreria thqual.SCSQLINK
- CSQ3ECMX nella libreria thqual.SCSQSNL x, dove x è la lettera della lingua:
  - thlqual.SCSQSNLE, per l'inglese americano con lettere maiuscole e minuscole
  - thlqual.SCSQSNLU, per l'inglese americano maiuscolo
  - thlqual.SCSQSNLK, per il giapponese
  - thlqual.SCSQSNLF, per il francese
  - thlqual.SCSQSNLC, per il cinese

IBM MQ include una modifica utente che sposta il contenuto della libreria thqual.SCSQSNL i nella libreria thqual.SCSQLINK e informa SMP/E. Questa modifica utente è denominata CSQ8UERL ed è descritta in *Program Directory for IBM MQ for z/OS* , per Long Term Support o Continuous Delivery. Per i collegamenti di download per le directory del programma, consultare [IBM MQ for z/OS Program Directory PDF files](#).

Una volta aggiornato il codice iniziale nelle librerie LPA, sarà disponibile dal successivo IPL z/OS (con l'opzione CLPA) a tutti i sottosistemi del gestore code aggiunti durante l'IPL dalle definizioni nei membri IEFSSNss in SYS1.PARMLIB.

È possibile renderla immediatamente disponibile senza un IPL per qualsiasi nuovo sottosistema del gestore code aggiunto successivamente (come descritto in [“Define the IBM MQ subsystem to z/OS”](#) a pagina 896 ) aggiungendolo all'LPA come segue:

- Se non è stato utilizzato CSQ8UERL, immettere i seguenti comandi z/OS :

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thqua1.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqua1.SCSQSNL x
```

- Se si utilizza CSQ8UERL, è possibile caricare il codice iniziale nell'area LPA utilizzando il comando z/OS riportato di seguito:

```
SETPROG LPA,ADD,MASK=*,DSNAME=thqua1.SCSQLINK
```

- Se si utilizza Advanced Message Security , è necessario immettere anche il seguente comando z/OS per includere un modulo aggiuntivo nell'area LPA:

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thqua1.SCSQLINK
```

Se è stata applicata la manutenzione o se si intende riavviare un gestore code con una versione o una release successive di IBM MQ, il codice precedente può essere reso disponibile per i gestori code esistenti utilizzando la seguente procedura. I gestori code per i quali non si eseguono queste operazioni

continuano a utilizzare la versione del codice iniziale già utilizzata. Non è necessario eseguire questa procedura per tutti i gestori code su una LPAR, a meno che non si stia specificamente tentando di applicare la manutenzione a tutti i gestori code o di aggiornarli tutti a una versione o release più recenti di IBM MQ.

1. Aggiungerlo all'LPA utilizzando i comandi z/OS SETPROG come descritto in precedenza in questo argomento.
2. Arrestare il gestore code utilizzando il comando IBM MQ STOP QMGR.
3. Accertarsi che qmgr.REFRESH.QMGR è impostato. Consultare [Comandi MQSC, profili e relativi livelli di accesso](#).
4. Aggiornare il codice iniziale per il gestore code utilizzando il comando IBM MQ REFRESH QMGR TYPE (EARLY).
5. Riavviare il gestore code utilizzando il comando IBM MQ START QMGR.

I comandi IBM MQ STOP QMGR, REFRESH QMGR e START QMGR sono descritti in [Comandi MQSC](#).

## Altro codice

Tutti i moduli di caricamento forniti da IBM MQ nelle seguenti librerie sono rientranti e possono essere inseriti nell'area LPA:


- SCSQAUTH
- SCSQANL x, dove x è la tua lettera della lingua
- SCSQMVR1

**Importante:** Tuttavia, se si posizionano le librerie nell'area LPA, ogni volta che si applica la manutenzione, è necessario copiare manualmente i moduli modificati nell'area LPA. Pertanto è preferibile mettere il IBM MQ caricare le librerie nell'elenco dei collegamenti, che possono essere aggiornate dopo la manutenzione emettendo il filez/OS comando MODIFICA LLA AGGIORNA.

Vedere [Modifica del contenuto dei set di dati LNKLST](#) per ulteriori informazioni e [Utilizzo della struttura dinamica LNKLST in modo sicuro e corretto](#).

Questo è particolarmente consigliato per SCSQAUTH in modo che non sia necessario includerlo in diversi STEPLIBs. Solo una libreria di lingua, SCSQANL x, deve essere inserita nell'elenco LPA o link. Le librerie dell'elenco dei collegamenti sono specificate in un membro LNKLSTkk di SYS1.PARMLIB.


La funzione di accodamento distribuito e CICS bridge (ma non il gestore code stesso) richiedono l'accesso alla libreria di runtime LE (Language Environment) SCEERUN. Se si utilizza una di queste funzioni, è necessario includere SCEERUN nell'elenco dei collegamenti.

 Alcuni moduli vengono caricati all'avvio del gestore code in ECSA. In ambienti vincolati ECSA, è possibile inserire questi moduli nell'area LPA. Consultare [“Placing IBM MQ global modules into the LPA”](#) a pagina 893.

## Concetti correlati

[“Update the z/OS program properties table”](#) a pagina 895

Some additional PPT entries are needed for the IBM MQ queue manager.

  *Placing IBM MQ global modules into the LPA*

When an IBM MQ for z/OS queue manager starts up, it loads some of its load modules (global modules) into the extended common service area (ECSA). At queue manager shut down the ECSA is freed.

There are 19 global modules, which at IBM MQ 9.3, consumed approximately 1.2 MB of ECSA for each running queue manager.

**Note:** Although CSQ7GPLM is a global module, it should not be added to the LPA.

In environments that run multiple queue managers for each LPAR, and require a reduction in ECSA consumption due to ECSA or high private constraints, it is possible to place the global modules into the LPA. Placing the global modules of IBM MQ into the LPA is a manual process that requires care, so

you should only carry out this procedure if there is a significant need to address ECSA or high private constraints.

If the queue manager cannot find a global module in its STEPLIB, and detects the module is in the LPA, it uses the LPA copy directly, instead of loading a copy of the module into ECSA. Alternatively if the queue managers code is normally loaded from the link list then any global modules in the LPA are loaded in preference to any global modules in the link list.

The z/OS common storage tracking function (see [Using the common storage tracking function](#)) tracks the storage under the each queue manager's MSTR address space for each queue manager and can be used to detect how much space is being use by the global modules.

By default, the global modules are in the SCSQAUTH load library. If the MSTR address space of a queue manager locates SCSQAUTH through the STEPLIB concatenation, the global modules from there are used in preference to any in the LPA and are loaded into ECSA.

The global modules are:

CSQ0GPLM, CSQ3AMGP, CSQ3SSGP, CSQ9PREP,  
CSQ9SCNB, CSQGGPLM, CSQMCGLM, CSQMGPLM, CSQRGLM1,  
CSQSLD1, CSQVGEPL, CSQVSRX, CSQWDL2, CSQWDL3,  
CSQWVZSA, CSQWZDGO, CSQWVZPS, CSQWVGTM, CSQZTDDM

### Important:

- The name of the global modules for IBM MQ remain constant across different IBM MQ versions. Therefore, if you load global modules into the LPA, they should be from a single IBM MQ version, and should only be used by queue managers running at the same IBM MQ version.
- If multiple versions of IBM MQ are run on the same LPAR then only one of those can have its global modules in the LPA at any given time.
- If maintenance is applied to an IBM MQ installation which has global modules loaded into the LPA, and that maintenance updates any of the global modules, you should perform the procedure described in the following text again.

## Procedure

To put the global modules from a version of IBM MQ into the LPA, perform the following steps:

1. Create a copy of the th1qua1 . SCSQAUTH load library, and its contents, for example: th1qua1 . LOCAL . SCSQAUTH. Ensure that this load library is protected from unauthorized access using your external security manager (ESM).
2. APF authorize the th1qua1 . LOCAL . SCSQAUTH load library; see [“LAPF autorizza le librerie di caricamento IBM MQ”](#) on page 890.
3. Create a new th1qua1 . GLOBAL . SCSQAUTH load library with the same attributes as th1qua1 . LOCAL . SCSQAUTH.

**Note:** This load library does not need to be APF authorized. Ensure that this load library is protected from unauthorized access using your ESM.

4. Copy the 19 global modules from th1qua1 . LOCAL . SCSQAUTH into th1qua1 . GLOBAL . SCSQAUTH.
5. Delete the 19 global modules from th1qua1 . LOCAL . SCSQAUTH.
6. Place the 19 global modules from th1qua1 . GLOBAL . SCSQAUTH into the LPA, by either:
  - a. Adding th1qua1 . GLOBAL . SCSQAUTH into an LPALSTxx member of SYS1.PARMLIB. You must then IPL the system with the CLPA option to ensure that the library contents are loaded into the PLPA.
  - b. Dynamically adding the modules to the LPA using the following command:

```
SETPROG
LPA,ADD,MODNAME=(CSQ0GPLM,CSQ3AMGP,CSQ3SSGP,CSQ9PREP,CSQ9SCNB,CSQGGPLM,
```

**Note:** LPA1STxx is the preferred long term means of placing modules in LPA.

7. Validate that the modules are in the LPA by issuing the following command:

```
D PROG,LPA,MODNAME=CSQMCGLM
```

The output of the command should indicate the entry and load points of the module if it was successfully loaded into the LPA.

For each queue manager that needs to use the global modules from the LPA, then if you normally place:

1. thlqual.SCSQAUTH in the link list, just stop and start your queue manager. The global modules are loaded from the LPA, and the local modules from the link list.
2. thlqual.SCSQAUTH in the MSTR JCL STEPLIB, change the JCL so that the STEPLIB uses thlqual.LOCAL.SCSQAUTH instead of thlqual.SCSQAUTH. Stop and start the queue manager; the global modules are loaded from the LPA, and the local modules from the STEPLIB.

The CHIN and AMSM JCL can continue to use thlqual.SCSQAUTH as can any IBM MQ applications.

To revert the queue manager to loading the global modules into ECSA perform the following steps:

1. Stop the queue managers
2. Remove the global modules from the LPA, either at the next IPL by removing the LPA1STxx definitions or by using the following command:

```
SETPROG LPA,DELETE,MODNAME=(xxx) FORCE=YES
```

3. If thlqual.LOCAL.SCSQAUTH is in the STEPLIB of the queue manager replace it with thlqual.SCSQAUTH.
4. Restart the queue managers.

### Related concepts

[“Aggiornare l'elenco di link z/OS e LPA” on page 891](#)

Aggiornare le librerie LPA con la nuova versione delle prime librerie di codice. L'altro codice può essere inserito nell'elenco dei link o nell'area LPA.

### **Update the z/OS program properties table**

Some additional PPT entries are needed for the IBM MQ queue manager.

- *You must perform this task once for each z/OS system where you want to run IBM MQ.*
- *If you are using queue sharing groups, you must ensure that the settings for IBM MQ are identical on each z/OS system in the sysplex.*
- *You do not need to perform this task when migrating from a previous version.*
- *You do need to perform the CSQ0DSRV part of this task when you require Advanced Message Security.*

A sample containing all the required PPT entries is provided in thlqual.SCSQPROC(CSQ4SCHD). Ensure that the required entries are added to the PPT, which you can find in SYS1.PARMLIB(SCHEDxx).

In z/OS, CSQYASCP is already defined to the operating system with the attributes detailed and no longer needs to be included in a SCHEDxx member of PARMLIB.

The IBM MQ queue manager controls swapping itself. However, if you have a heavily-loaded IBM MQ network and response time is critical, it might be advantageous to make the IBM MQ channel initiator nonswappable, by adding the CSQXJST PPT entry, at the risk of affecting the performance of the rest of your z/OS system.

If you require Advanced Message Security, add the CSQ0DSRV PPT entry.

Issue the z/OS command **SET SCH=xx**, where xx is the suffix of the SCHEDxx member of PARMLIB, for these changes to take effect.

## Related concepts

[“Define the IBM MQ subsystem to z/OS” on page 896](#)

Update the subsystem name table and decide on a convention for command prefix strings.

## **Configuring the queue manager and channel initiator**

Use these topics as a step by step guide for configuring the queue manager and channel initiator.

## **Define the IBM MQ subsystem to z/OS**

Update the subsystem name table and decide on a convention for command prefix strings.

Repeat this task for each IBM MQ queue manager. You do not need to perform this task when migrating from a previous version.

## Related concepts

[“Create procedures for the IBM MQ queue manager” on page 899](#)

Each IBM MQ subsystem needs a cataloged procedure to start the queue manager. You can create your own or use the IBM-supplied procedure library.

## **Updating the subsystem name table**

When defining the IBM MQ subsystem you must add an entry to the subsystem name table.

The subsystem name table of z/OS, which is taken initially from the SYS1.PARMLIB member IEFSSNss, contains the definitions of formally defined z/OS subsystems. To define each IBM MQ subsystem, you must add an entry to this table, either by changing the IEFSSNss member of SYS1.PARMLIB, or, preferably, by using the z/OS command SETSSI.

IBM MQ subsystem initialization supports parallel processing, so IBM MQ subsystem definition statements can be added both above and below the BEGINPARALLEL keyword in the IEFSSNss table available at z/OS V1.12 and later.

If you use the SETSSI command, the change takes effect immediately, and there is no need to perform an IPL of your system. Ensure you update SYS1.PARMLIB as well, as described in [“Update SYS1.PARMLIB members” on page 903](#) so that the changes remain in effect after subsequent IPLs.

The SETSSI command to dynamically define an IBM MQ subsystem is:

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

The corresponding information in IEFSSNss can be specified in one of two ways:

- The keyword parameter form of the IBM MQ subsystem definition in IEFSSNss. This is the recommended method.

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- The positional parameter form of the IBM MQ subsystem definition.

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

Do not mix the two forms in one IEFSSNss member. If different forms are required, use a separate IEFSSNss member for each type, adding the SSN operand of the new member to the IEASYSpp SYS1.PARMLIB member. To specify more than one SSN, use SSN=(aa,bb,...) in IEASYSpp.

In the examples,



## ssid

The subsystem identifier. It can be up to four characters long. All characters must be alphanumeric (uppercase A through Z, 0 through 9), it must start with an alphabetic character. The queue manager will have the same name as the subsystem, therefore you can use only characters that are allowed for both z/OS subsystem names and IBM MQ object names.

## cpf

The command prefix string (see [“Defining command prefix strings \(CPFs\)”](#) on page 897 for information about CPFs).

## scope

The system scope, used if you are running in a z/OS sysplex (see [“CPFs in a sysplex environment”](#) on page 898 for information about system scope).

Figure 97 on page 897 shows several examples of IEFSSNss statements.

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

Figure 97. Sample IEFSSNss statements for defining subsystems

**Note:** When you have created objects in a subsystem, you cannot change the subsystem name or use the page sets from one subsystem in another subsystem. To do either of these, you must unload all the objects and messages from one subsystem and reload them into another.

Table 55 on page 897 gives a number of examples showing the associations of subsystem names and command prefix strings (CPFs), as defined by the statements in [Figure 97 on page 897](#).

IBM MQ subsystem name	CPF
CSQ1	+mqs1cpf
CSQ2	+mqs2cpf
CSQ3	++

**Note:** The ACTIVATE and DEACTIVATE functions of the z/OS command SETSSI are not supported by IBM MQ.

To check the status of the changes, issue the following command in SDSF: /D SSI,L. You will see the new subsystems created with ACTIVE status.

## Defining command prefix strings (CPFs)

Each subsystem instance of IBM MQ can have a command prefix string to identify that subsystem.

Adopt a system-wide convention for your CPFs for all subsystems to avoid conflicts. Adhere to the following guidelines:

- Define a CPF as string of up to eight characters.
- Do not use a CPF that is already in use by any other subsystem, and avoid using the JES backspace character defined on your system as the first character of your string.
- Define your CPF using characters from the set of valid characters listed in [Table 57 on page 898](#).
- Do not use a CPF that is an abbreviation for an already defined process or that might be confused with command syntax. For example, a CPF such as 'D' conflicts with z/OS commands such as DISPLAY. To avoid this happening, use one of the special characters (shown in [Table 57 on page 898](#)) as the first or only character in your CPF string.

- Do not define a CPF that is either a subset or a superset of an existing CPF. For an example, see [Table 56 on page 898](#).

*Table 56. Example of CPF subset and superset rules*

Subsystem name	CPF defined	Commands routed to
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1
MQC2	!C2	MQC2
MQB1	!B1	MQB

Commands intended for subsystem MQB1 (using CPF !B1) are routed to subsystem MQB because the CPF for this subsystem is !B, a subset of !B1. For example, if you entered the command:

```
!B1 START QMGR
```

subsystem MQB receives the command:

```
1 START QMGR
```

(which, in this case, it cannot deal with).

You can see which prefixes exist by issuing the z/OS command DISPLAY OPDATA.

If you are running in a sysplex, z/OS diagnoses any conflicts of this type at the time of CPF registration (see “[CPFs in a sysplex environment](#)” on page 898 for information about CPF registration).

[Table 57 on page 898](#) shows the characters that you can use when defining your CPF strings:


*Table 57. Valid character set for CPF strings*

Character set	Contents
Alphabetic	Uppercase A through Z, lowercase a through z
Numeric	0 through 9
National (see note)	@ \$ # (Characters that can be represented as hexadecimal values)
Special	. [] () * & + - = € <   ! ; % _ ? : >

**Note:**

The system recognizes the following hexadecimal representations of the national characters: @ as X'7C', \$ as X'5B', and # as X'7B'. In countries other than the U.S., the U.S. national characters represented on terminal keyboards might generate a different hexadecimal representation and cause an error. For example, in some countries the \$ character might generate an X'4A'.

The semicolon (;) is valid as a CPF but on most systems, this character is the command delimiter.

 **z/OS** *CPFs in a sysplex environment*

Use this topic to understand how to use CPFs within the scope of a sysplex.

If used in a sysplex environment, IBM MQ registers your CPFs to enable you to enter a command from any console in the sysplex and route that command to the appropriate system for execution. The command responses are returned to the originating console.

## Defining the scope for sysplex operation

Scope is used to determine the type of CPF registration performed by the IBM MQ subsystem when you are running IBM MQ in a sysplex environment.

Possible values for scope are as follows:

### M

System scope.

The CPF is registered with z/OS at system IPL time by IBM MQ and remains registered for the entire time that the z/OS system is active.

IBM MQ commands must be entered at a console connected to the z/OS image running the target subsystem, or you must use ROUTE commands to direct the command to that image.

Use this option if you are not running in a sysplex.

### S

Sysplex started scope.

The CPF is registered with z/OS when the IBM MQ subsystem is started, and remains active until the IBM MQ subsystem terminates.

You must use ROUTE commands to direct the original START QMGR command to the target system, but all further IBM MQ commands can be entered at any console connected to the sysplex, and are routed to the target system automatically.

After IBM MQ termination, you must use the ROUTE commands to direct subsequent START commands to the target IBM MQ subsystem.

### X

Sysplex IPL scope.

The CPF is registered with z/OS at system IPL time by IBM MQ and remains registered for the entire time that the z/OS system is active.

IBM MQ commands can be entered at any console connected to the sysplex, and are routed to the image that is executing the target system automatically.

An IBM MQ subsystem with a CPF with scope of S can be defined on one or more z/OS images within a sysplex, so these images can share a single subsystem name table. However, you must ensure that the initial START command is issued on (or routed to) the z/OS image on which you want the IBM MQ subsystem to run. If you use this option, you can stop the IBM MQ subsystem and restart it on a different z/OS image within the sysplex without having to change the subsystem name table or perform an IPL of a z/OS system.

An IBM MQ subsystem with a CPF with scope of X can only be defined on one z/OS image within a sysplex. If you use this option, you must define a unique subsystem name table for each z/OS image requiring IBM MQ subsystems with CPFs of scope X.

If you want to use the z/OS automatic restart manager (ARM) to restart queue managers in different z/OS images automatically, every queue manager must be defined in each z/OS image on which that queue manager might be restarted. Every queue manager must be defined with a sysplex-wide, unique 4-character subsystem name with a CPF scope of S.

### **Create procedures for the IBM MQ queue manager**

Each IBM MQ subsystem needs a cataloged procedure to start the queue manager. You can create your own or use the IBM-supplied procedure library.

- Repeat this task for each IBM MQ queue manager.
- You might need to modify the cataloged procedure when migrating from a previous version.

For each IBM MQ subsystem defined in the subsystem name table, create a cataloged procedure in a procedure library for starting the queue manager. The IBM-supplied procedure library is called SYS1.PROCLIB, but your installation might use its own naming convention.

The name of the queue manager started task procedure is formed by concatenating the subsystem name with the characters MSTR. For example, subsystem CSQ1 has the procedure name CSQ1MSTR. You need one procedure for each subsystem you define.

You need to include the library containing messages in your selected language:

- thlqual.SCSQSNLE, for US English mixed case
- thlqual.SCSQSNLU, for US English uppercase
- thlqual.SCSQSNLK, for Japanese
- thlqual.SCSQSNLF, for French
- thlqual.SCSQSNTL, for Chinese

Many examples and instructions in this product documentation assume that you have a subsystem called CSQ1. You might find these examples easier to use if a subsystem called CSQ1 is created initially for installation verification and testing purposes.

Two sample started task procedures are provided in thlqual.SCSQPROC. Member CSQ4MSTR uses one page set for each class of message, member CSQ4MSRR uses multiple page sets for the major classes of message. Copy one of these procedures to member xxxxMSTR (where xxxx is the name of your IBM MQ subsystem) of your SYS1.PROCLIB or, if you are not using SYS1.PROCLIB, your procedure library. Copy the sample procedure to a member in your procedure library for each IBM MQ subsystem that you define.

When you have copied the members, you can tailor them to the requirements of each subsystem, using the instructions in the member. For information about specifying limits of storage used by the queue manager, see [Storage configuration](#). You can also use symbolic parameters in the JCL to allow the procedure to be modified when it is started. If you have several IBM MQ subsystems, you might find it advantageous to use JCL include groups for the common parts of the procedure, to simplify future maintenance.

If you are using queue sharing groups, the STEPLIB concatenation must include the Db2 runtime target library SDSNLOAD, and it must be APF-authorized. This library is only required in the STEPLIB concatenation if it is not accessible through the link list or LPA.

#### Notes:

1. You can make a note of the names of your bootstrap data set (BSDS), logs, and page sets for use in JCL and then define these sets at a later step in the process.
2. Sample started task procedures CSQ4MSTR and CSQ4MSRR have been updated to include, but leave commented out, the CSQMINI DD card that can be used to define a QMINI data set that contains transport security, that is, SSL or TLS properties.

You can use [“The QMINI data set” on page 906](#) to enable or disable TLS 1.3 support and/or be used to define a custom list of CipherSpecs to be used by channels.

#### Related concepts

[“Create procedures for the channel initiator” on page 900](#)

For each IBM MQ subsystem, tailor a copy of CSQ4CHIN. Depending on what other products you are using, you might need to allow access to other data sets.

#### **Create procedures for the channel initiator**

For each IBM MQ subsystem, tailor a copy of CSQ4CHIN. Depending on what other products you are using, you might need to allow access to other data sets.

- Repeat this task for each IBM MQ queue manager.
- You might need to modify the cataloged procedure when migrating from a previous version.


You need to create a channel-initiator started task procedure for each IBM MQ subsystem that is going to use distributed queuing.

To do this:

1. Copy the sample started task procedure `thlqual.SCSQPROC(CSQ4CHIN)` to your procedure library. Name the procedure `xxxxx CHIN`, where `xxxxx` is the name of your IBM MQ subsystem (for example, `CSQ1CHIN` would be the channel initiator started task procedure for queue manager `CSQ1`).
2. Make a copy for each IBM MQ subsystem that you are going to use.
3. Tailor the procedures to your requirements using the instructions in the sample procedure `CSQ4CHIN`. You can also use symbolic parameters in the JCL to allow the procedure to be modified when it is started. This is described with the start options in [Administering IBM MQ for z/OS](#).

Concatenate the distributed queuing library `thlqual.SCSQMVR1`.

Access to the LE runtime library `SCEERUN` is required; if it is not in your link list (`SYS1.PARMLIB(LNKLSTkk)`), concatenate it in the `STEPLIB DD` statement.

 Consider adjusting the `MEMLIMIT` parameter using the information in [Storage configuration](#).

4. Authorize the procedures to run under your external security manager.
5. You need to include the library containing messages in your selected language:
  - `thlqual.SCSQSNLE`, for US English mixed case
  - `thlqual.SCSQSNLU`, for US English uppercase
  - `thlqual.SCSQSNLK`, for Japanese
  - `thlqual.SCSQSNLF`, for French
  - `thlqual.SCSQSNLC`, for Chinese

The channel initiator is a long running address space. To prevent its termination after a restricted amount of CPU has been consumed, confirm that either:

- The default for started tasks in your z/OS system is unlimited CPU; a JES2 configuration statement for `JOBCLASS(STC)` with `TIME=(1440,00)` achieves this, or
- Explicitly add a `TIME=1440`, or `TIME=NOLIMIT`, parameter to the `EXEC` statement for `CSQXJST`.

You can add the exit library (`CSQXLIB`) to this procedure later if you want to use channel exits. You need to stop and restart your channel initiator to do this.

If you are using TLS, access to the system TLS runtime library is required. This library is called `SIEALNKE`. The library must be APF authorized.

If you are using TCP/IP, the channel initiator address space must be able to access the `TCPIP.DATA` data set that contains TCP/IP system parameters. The ways that the data set has to be set up depends on which TCP/IP product and interface you are using. They include:

- Environment variable, `RESOLVER_CONFIG`
- `/etc/resolv.conf` on the file system
- `//SYSTCPD DD` statement
- `//SYSTCPDD DD` statement
- `jobname/userid.TCPIP.DATA`
- `SYS1.TCPPARMS(TCPDATA)`
- `zqpname.TCPIP.DATA`

Some of these affect your started-task procedure JCL. For more information, see [z/OS Communications Server: IP Configuration Guide](#).

### Related concepts

[“Define the IBM MQ subsystem to a z/OS WLM service class” on page 902](#)

To give IBM MQ appropriate performance priority in the z/OS system, you must assign the queue manager and channel initiator address spaces to an appropriate z/OS workload management (WLM) service class. If you do not do this explicitly, inappropriate defaults might apply.

### **Define the IBM MQ subsystem to a z/OS WLM service class**

To give IBM MQ appropriate performance priority in the z/OS system, you must assign the queue manager and channel initiator address spaces to an appropriate z/OS workload management (WLM) service class. If you do not do this explicitly, inappropriate defaults might apply.

- Repeat this task for each IBM MQ queue manager.
- You do not need to perform this task when migrating from a previous version.

Use the ISPF dialog supplied with WLM to perform the following tasks:

- Extract the z/OS WLM policy definition from the WLM couple data set.
- Update this policy definition by adding queue manager and channel initiator started task procedure names to the chosen service class
- Install the changed policy on the WLM couple data set

Then activate this policy using the z/OS command

```
V WLM,POLICY=policyname,REFRESH
```

See [Planning your IBM MQ environment on z/OS](#) for more information on setting performance options.

#### **Related concepts**

[“Set up the Db2 environment” on page 939](#)

If you are using queue sharing groups you must create the required Db2 objects by customizing and running a number of sample jobs.

### **Implement your ESM security controls**

Implement security controls for queue managers and the channel initiator.

- Repeat this task for each IBM MQ queue manager.
- You might need to perform this task when migrating from a previous version.

If you use RACF® as your external security manager, see [Setting up security on z/OS](#), which describes how to implement these security controls.

If you are using the channel initiator, you must also do the following:

- If your subsystem has connection security active, define a connection security profile ssid.CHIN to your external security manager (see [Connection security profiles for the channel initiator](#) for information about this).
- If you are using Transport Layer Security (TLS) or a sockets interface, ensure that the user ID under whose authority the channel initiator is running is configured to use z/OS UNIX System Services, as described in the [z/OS UNIX System Services Planning](#) documentation.
- If you are using TLS, ensure that the user ID under whose authority the channel initiator is running is configured to access the key ring specified in the SSLKEYR parameter of the ALTER QMGR command.

Before you start the queue manager, set up IBM MQ data set and system security by:

- Authorizing the queue manager started task procedure to run under your external security manager.
- Authorizing access to the queue manager data sets.
- Configuring z/OS data set encryption if required.

See the section, [confidentiality for data at rest on IBM MQ for z/OS with data set encryption](#), for more information.

For details about how to do this, see [Security installation tasks for z/OS](#).

If you are using RACF, provided you use the RACF STARTED class, you do not need to perform an IPL of your system (see [RACF authorization of started-task procedures](#)).

### Related concepts

[“Update SYS1.PARMLIB members” on page 903](#)

To ensure that your changes remain in effect after an IPL, you must update some members of SYS1.PARMLIB

[“Implement ESM security controls for the queue sharing group” on page 942](#)

Implement security controls for all queue managers in a queue sharing group, to access Db2 and the coupling facility list structures.

### **Update SYS1.PARMLIB members**

To ensure that your changes remain in effect after an IPL, you must update some members of SYS1.PARMLIB

- *You need to perform this task once for each z/OS system where you want to run IBM MQ.*
- *If you are using queue sharing groups, you must ensure that the settings for IBM MQ are identical on each z/OS system in the sysplex.*
- *You might need to perform this task when migrating from a previous version.*

Update SYS1.PARMLIB members as follows:

1. Update member IEFSSNss as described in [“Define the IBM MQ subsystem to z/OS” on page 896](#).
2. Change IEASYSpp so that the following members are used when an IPL is performed:
  - the PROGxx or IEAAPFaa members used in [“L'APF autorizza le librerie di caricamento IBM MQ” on page 890](#)
  - the LNKLSTkk and LPALSTmm members used in [“Aggiornare l'elenco di link z/OS e LPA” on page 891](#)
  - the SCHEDxx member used in [“Update the z/OS program properties table” on page 895](#)
  - the IEFSSNss member used in [“Define the IBM MQ subsystem to z/OS” on page 896](#)

### Related concepts

[“Customize the initialization input data sets” on page 903](#)

Make working copies of the sample initialization input data sets and tailor them to suit your system requirements.

### **Customize the initialization input data sets**

Make working copies of the sample initialization input data sets and tailor them to suit your system requirements.

- *Repeat this task for each IBM MQ queue manager.*
- *You need to perform this task when migrating from a previous version.*

Each IBM MQ queue manager gets its initial definitions from a series of commands contained in the IBM MQ *initialization input data sets*. These data sets are referenced by the DD names CSQINP1, CSQINP2, and CSQINPT defined in the queue manager started task procedure.

Responses to these commands are written to the initialization output data sets referenced by the DD names CSQOUT1, CSQOUT2 and CSQOUTT.

To preserve the originals, make working copies of each sample. Then you can tailor the commands in these working copies to suit your system requirements.

If you use more than one IBM MQ subsystem, if you include the subsystem name in the high-level qualifier of the initialization input data set name, you can identify the IBM MQ subsystem associated with each data set more easily.

Refer to the following topics for further information about the samples:

- [Initialization data set formats](#)
- [Using the CSQINP1 sample](#)
- [Using the CSQINP2 samples](#)
- [Using the CSQINPX sample](#)
- [Using the CSQINPT sample](#)

## Initialization data set formats

The initialization input data sets can be partitioned data set (PDS) members or sequential data sets. They can be a concatenated series of data sets. Define them with a record length of 80 bytes, where:

- Only columns 1 through 72 are significant. Columns 73 through 80 are ignored.
- Records with an asterisk (\*) in column 1 are interpreted as comments and are ignored.
- Blank records are ignored.
- Each command must start on a new record.
- A trailing - means continue from column 1 of the next record.
- A trailing + means continue from the first non-blank column of the next record.
- The maximum number of characters permitted in a command is 32 762.

The initialization output data sets are sequential data sets, with a record length of 125, a record format of VBA, and a block size of 629.

## Using the CSQINP1 sample

Data set th1qua1 .SCSQPROC holds two members which contain definitions of buffer pools, page set to buffer pool associations, and an ALTER SECURITY command.

Member CSQ4INP1 uses one page set for each class of message. The messages are divided into the following classes:

- System-related messages.
- Important long-lived messages.
- Short-lived messages.
- Miscellaneous messages.

Member CSQ4INPR uses multiple page sets for each of the major classes of message, and one page set for each other class. The following are the major classes of messages:

- Important long-lived messages.
- Short-lived messages.

Include the appropriate sample in the CSQINP1 concatenation of your queue manager started task procedure.

### Notes:

1. IBM MQ supports up to 100 buffer pools in the range zero through 99. The DEFINE BUFFPOOL command can only be issued from a CSQINP1 initialization data set. The definitions in the sample specify four buffer pools.
2. Each page set used by the queue manager must be defined in the CSQINP1 initialization data set by using the DEFINE PSID command. The page set definition associates a buffer pool ID with a page set. If no buffer pool is specified, buffer pool zero is used by default.  
  
Page set zero (00) must be defined. It contains all the object definitions. You can define up to 100 page sets for each queue manager.
3. The ALTER SECURITY command can be used to alter the security attributes TIMEOUT and INTERVAL. In CSQ4INP1, the default values are defined as 54 for TIMEOUT and 12 for INTERVAL.



See [Planning your page sets and buffer pools](#) for information about organizing buffer pools and page sets.

If you change the buffer pool and page set definitions dynamically while the queue manager is running, you should also update the CSQINP1 definitions. The changes are only retained for a cold start of IBM MQ, unless the buffer pool definition includes the REPLACE attribute.

## Using the CSQINP2 samples

This table lists the members of `thlqual.SCSQPROC` that can be included in the CSQINP2 concatenation of your queue manager started task procedure, with a description of their function. The naming convention is CSQ4IN\*. CSQ4INY\* members should be modified for your configuration. You should avoid changing CSQINS\* members because you will need to reapply any changes when you migrate to the next release. Instead, you can put DEFINE or ALTER commands in CSQ4INY\* members.

<i>Table 58. Members of thlqual.SCSQPROC</i>	
Member name	Description
CSQ4INSG	System object definitions.
CSQ4INSA	System object and default rules for channel authentication.
CSQ4INSX	System object definitions.
CSQ4INSS	Customize and include this member if you are using queue sharing groups.
CSQ4INSJ	Customize and include this member if you are using publish/subscribe using JMS.
CSQ4INSM	System object definitions for Advanced Message Security.
CSQ4INSR	Customize and include this member if you are using WebSphere Application Server, or the queued publish/subscribe interface supported by the queued publish/subscribe daemon in IBM MQ.
CSQ4DISP	CSQINP2 sample for displaying object definitions.
CSQ4INYC	Clustering definitions.
CSQ4INYD	Distributed queuing definitions.
CSQ4INYG	General definitions.
CSQ4INYR	Storage class definitions, using multiple page sets for the major classes of message.
CSQ4INYS	Storage class definitions, using one page set for each class of message.

You need to define objects once only, not each time that you start a queue manager, so it is not necessary to include these definitions in CSQINP2 every time. If you do include them every time, you are attempting to define objects that already exist, and you will get messages similar to the following:

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL' ABNORMAL COMPLETION
```

The objects are not damaged by this failure. If you want to leave the SYSTEM definitions data set in the CSQINP2 concatenation, you can avoid the failure messages by specifying the REPLACE attribute against each object.

## Using the CSQINPX sample

Sample `thlqual.SCSQPROC(CSQ4INPX)` contains a set of commands that you might want to execute each time the channel initiator starts. These are typically channel-related commands such as START LISTENER, which are required every time the channel initiator starts, rather than whenever the queue manager starts, and which are not allowed in the input data sets CSQINP1 or CSQINP2. You must

customize this sample before use; you can then include it in the CSQINPX data set for the channel initiator.

The IBM MQ commands contained in the data set are executed at the end of channel initiator initialization, and output is written to the data set specified by the CSQOUTX DD statement. The output is like that produced by the COMMAND function of the IBM MQ utility program (CSQUTIL). See [Using the CSQUTIL utility for IBM MQ for z/OS](#).

You can specify any of the IBM MQ commands that can be issued from CSQUTIL, not only the channel commands. You can enter commands from other sources while CSQINPX is being processed. All commands are issued in sequence, regardless of the success of the previous command.

To specify a command response time, you can use the pseudo-command COMMAND as the first command in the data set. This takes a single optional keyword RESPTIME( *nnn* ), where *nnn* is the time, in seconds, to wait for a response to each command. This is in the range 5 through 999; the default is 30.

If IBM MQ detects that the responses to four commands have taken too long, processing of CSQINPX is stopped and no further commands are issued. The channel initiator is not stopped, but message CSQU052E is written to the CSQOUTX data set, and message CSQU013E is sent to the console.

When IBM MQ has completed processing of CSQINPX successfully, message CSQU012I is sent to the console.

## Using the CSQINPT sample

This table lists the members of thlqual . SCSQPROC that can be included in the CSQINPT concatenation of your queue manager started task procedure, with a description of their function.

<i>Table 59. Members of thlqual . SCSQPROC</i>	
Member name	Description
CSQ4INST	System default subscription definition.
CSQ4INYT	Publish/Subscribe definitions.

The IBM MQ commands contained in the data set are executed when publish/subscribe initialization completes, and output is written to the data set specified by the CSQOUTT DD statement. The output is like that produced by the COMMAND function of the IBM MQ utility program (CSQUTIL). See [Using the CSQUTIL utility for IBM MQ for z/OS](#).

### Related concepts

[“Create the bootstrap and log data sets” on page 908](#)

Use the supplied program CSQJU003 to prepare the bootstrap data sets (BSDSs) and log data sets.

### The QMINI data set

You can use the QMINI data set to specify properties that are to be read and processed during queue manager initialization.

## Characteristics of the QMINI data set

The QMINI data set is a sequential data set, or a member of a partitioned data set, with a maximum record length of 80 bytes (72 bytes for data and eight bytes for the line number).

The following example shows the properties for a sequential QMINI data set. Some properties are, of course, based on your environment.

```
Data Set Name : QM01.QMINI
General Data
Management class . . : STANDARD Current Allocation
Storage class : STANDARD Allocated tracks . : 1
Volume serial : P5P21E Allocated extents . : 1
Device type : 3390
```

```

Data class : **None**
Organization : PS Current Utilization
Record format : FB Used tracks : 0
Record length : 80 Used extents : 0
Block size : 3120
1st extent tracks . . : 1
Secondary tracks . . . : 1 Dates
Data set name type . . : Creation date : 2020/08/11
Data set encryption . : NO Referenced date . . . : ***None***
SMS Compressible . . . : NO Expiration date . . . : ***None***

```

thlqua1.SCSQPROC, includes:

- The sample contents for a QMINI data set in CSQ4QMIN.
- An example of specifying the QMINI data set using the //CSQMINI DD card, in the queue manager startup JCL, in the started task procedures CSQ4MSTR and CSQ4MSRR.

#### Notes:

- The code that parses the data set only parses the first 72 bytes of each record.
- Line numbers are ignored so it is not necessary to specify line numbers.
- If a line starts with an asterisk character (\*), the line is treated as a comment.
- The contents of the QMINI data set are parsed during queue manager startup. If the contents are parsed successfully, message CSQM578I is issued in the queue manager job log. If any errors are encountered during parsing, error messages, for example [CSQM573E](#), are issued in the queue manager job log but the queue manager still starts.

Check for error messages, and resolve any issues in the contents of the QMINI data set.

If the queue manager is unable to parse the QMINI data set, you can start the channel initiator, but you cannot start any channels that are configured to use SSL or TLS as the security configuration settings are unknown.

- If you make any updates to the data set after you have started the queue manager, you must restart the queue manager to pick up the changes.

## The TransportSecurity stanza

From IBM MQ for z/OS 9.2.0, the QMINI data set supports the TransportSecurity stanza. This stanza provides similar function to that provided by the SSL stanza in the qm.ini file on IBM MQ for Multiplatforms.

The TransportSecurity stanza supports the following properties:

#### AllowTLSV13

Whether a queue manager is able to use the TLS 1.3 CipherSpecs; valid values are: *TRUE/T/YES/Y* or *FALSE/F/NO/N*.

For migrated queue managers, TLS 1.3 is not enabled by default. You can enable TLS 1.3 by defining a QMINI data set with the TransportSecurity stanza and **AllowTLSV13=TRUE**.

For newly created queue managers TLS 1.3 is enabled by default.

#### AllowedCipherSpecs

A custom list of CipherSpecs that are enabled.

See [Providing a custom list of ordered and enabled CipherSpecs on IBM MQ for z/OS](#) for more information on this property.

Duplicate CipherSpec names in the list are ignored.

#### OutboundSNI

Whether the Server Name Indication (SNI) is set to the target IBM MQ channel name to the remote system when initiating a TLS connection, or to the hostname; valid values are: CHANNEL or HOSTNAME.

If the destination channel is configured with a certificate label on the channel object CERTLABL field, you must set CERTLABL to the channel value. If a connection with a setting of HOSTNAME is made to

a channel with a CERTLABL setting, the connection fails and an AMQ9673 message is printed in the remote queue manager error logs.

The following example shows how the TransportSecurity stanza is specified:

```
TransportSecurity:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
 ECDHE_RSA_AES_256_GCM_SHA384
AllowTLSV13=TRUE
```

## **Create the bootstrap and log data sets**

Use the supplied program CSQJU003 to prepare the bootstrap data sets (BSDSs) and log data sets.

### **Note:**

- Repeat this task for each IBM MQ queue manager.
- If you are using z/OS data set encryption to protect the BSDS or active log data sets, you must configure this option before the data sets are allocated in this step.
- You do not need to perform this task when migrating from a previous version.
- If you are migrating a queue manager and adding z/OS data set encryption for active log data sets or BSDS, you need to convert the data sets.
- For more information about configuring z/OS data set encryption, and converting existing IBM MQ data sets to be encrypted, see [confidentiality for data at rest on IBM MQ for z/OS with data set encryption..](#)

The sample JCL and Access Method Services (AMS) control statements to run CSQJU003 to create a single or dual logging environment are held in thlqual.SCSQPROC(CSQ4BSDS). Customize and run this job to create your BSDSs and logs and to preformat the logs.

**Important:** You should use the newest version of CSQ4BSDS, or update your JCL manually to use RECORDS(850 60).

The started task procedure, CSQ4MSTR, described in [“Create procedures for the IBM MQ queue manager”](#) on page 899, refers to BSDSs in statements of the form:

```
//BSDS1 DD DSN=++HLQ++.BSDS01,DISP=SHR
//BSDS2 DD DSN=++HLQ++.BSDS02,DISP=SHR
```

The log data sets are referred to by the BSDSs.

### **Note:**

1. The BLKSIZE must be specified on the SYSPRINT DD statement in the LOGDEF step. The BLKSIZE must be 629.
2. To help identify bootstrap data sets and log data sets from different queue managers, include the subsystem name in the high level qualifier of these data sets.
3. If you are using queue sharing groups, you must define the bootstrap and log data sets with SHAREOPTIONS(2 3).

See [Pianificazione su z/OS](#) for information about planning bootstrap and log data sets and their sizes.

From IBM MQ 8.0, the 8 byte log RBA enhancement improves the availability of a queue manager, as described in [Larger log Relative Byte Address](#). To enable 8 byte log RBA on a queue manager before the queue manager is first started, perform the following steps after creating your logging environment.

**Note:** For queue managers created at IBM MQ 9.3.0 or later 8 byte log RBA is already enabled, so the following steps are not necessary.

1. Using **IDCAMS ALTER**, rename the version 1 format BSDSs (created using the CSQJU003 program) to something like ++HLQ++. V1 .BSDS01.

**Note:** Ensure that you rename the data and index components as well as the VSAM cluster.

2. Allocate new BSDSs with the same attributes as the ones already defined. These will become the version 2 format BSDSs that will be used by the queue manager when it is started.
3. Run the BSDS conversion utility (CSQJUCNV) to convert the version 1 format BSDSs to the new version 2 format BSDSs.
4. Once the conversion completes successfully, delete the version 1 format BSDSs.

**Note:** If the queue manager is in a queue sharing group, all queue managers in the queue sharing group must have been started as follows before 8 byte log RBA can be enabled:

- If the queue manager is at IBM MQ 9.0.0 LTS it must have been started with **OPMODE(NEWFUNC,900)** or **OPMODE(NEWFUNC,800)**
- If the queue manager is at IBM MQ 9.0.n CD, or IBM MQ 9.1.0 LTS, or later, it needs to have been started at that level

### Related concepts

[“Define your page sets” on page 909](#)

Define page sets for each queue manager using one of the supplied samples.

### **Define your page sets**

Define page sets for each queue manager using one of the supplied samples.

- *Repeat this task for each IBM MQ queue manager.*

*If you using z/OS data set encryption to protect the page sets, you must configure this option before the data sets are allocated in this step.*

- *You do not need to perform this task when migrating from a previous version.*

*If you are migrating a queue manager and adding z/OS data set encryption for page sets, you need to convert the page sets.*

*See the section, [Confidentiality for data at rest on IBM MQ for z/OS with data set encryption](#). for more information about configuring z/OS data set encryption and converting existing IBM MQ data sets to be encrypted.*

Define separate page sets for each IBM MQ queue manager. thlqual.SCSQPROC(CSQ4PAGE) and thlqual.SCSQPROC(CSQ4PAGR) contain JCL and z/OS access method services (AMS) control statements to define and format page sets. Member CSQ4PAGE uses one page set for each class of message, member CSQ4PAGR uses multiple page sets for the major classes of message. The JCL runs the supplied utility program CSQUTIL. Review the samples and customize them for the number of page sets you want and the sizes to use. See [Planning your page sets and buffer pools](#) for information about page sets and how to calculate suitable sizes.

The started task procedure CSQ4MSTR described in [“Create procedures for the IBM MQ queue manager” on page 899](#) refers to the page sets, in a statement of the form:

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxx
```

where *nn* is the page set number between 00 and 99, and *xxxxxxxx* is the data set that you define.

### Note:

1. If you intend to use the dynamic page set expansion feature, ensure that secondary extents are defined for each page set. thlqual.SCSQPROC(CSQ4PAGE) shows how to do this.
2. To help identify page sets from different queue managers, include the subsystem name in the high level qualifier of the data set associated with each page set.
3. If you intend to allow the FORCE option to be used with the FORMAT function of the utility program CSQUTIL, you must add the REUSE attribute on the AMS DEFINE CLUSTER statement.

See the [Optional Parameters](#) section of the z/OS DEFINE CLUSTER command for more information on REUSE.

4. If your page sets are to be larger than 4 GB you must use the Storage Management System (SMS) EXTENDED ADDRESSABILITY function.

### Related concepts

[“Add the IBM MQ entries to the Db2 tables” on page 942](#)

If you are using queue sharing groups, run the CSQ5PQSG utility to add queue sharing group and queue manager entries to the IBM MQ tables in the Db2 data-sharing group.

### Tailor your system parameter module

The IBM MQ system parameter module controls the logging, archiving, tracing, and connection environments that IBM MQ uses in its operation. A default module is supplied. You should create your own system parameter module as some parameters, for example data set names, are usually site specific.

- Repeat this task for each IBM MQ queue manager, as required.
- You might need to perform this task when migrating from a previous version. For details, see [Migrating IBM MQ on z/OS](#).
- To enable Advanced Message Security for z/OS on an existing queue manager, you only need to set SPLCAP to YES as described in [“Utilizzo di CSQ6SYSP” on page 912](#). If you are configuring this queue manager for the first time, complete the whole of this task.

The system parameter module has four macros as follows:

Macro name	Purpose
CSQ6SYSP	Specifies the connection and tracing parameters, see <a href="#">“Utilizzo di CSQ6SYSP” on page 912</a>
CSQ6LOGP	Controls log initialization, see <a href="#">“Using CSQ6LOGP” on page 921</a>
CSQ6ARVP	Controls archive initialization, see <a href="#">“Using CSQ6ARVP” on page 926</a>
CSQ6USGP	Controls usage recording, see <a href="#">“Using CSQ6USGP” on page 932</a>

IBM MQ supplies a default system parameter module, CSQZPARM, which is invoked automatically if you issue the START QMGR command (without a PARM parameter) to start an instance of IBM MQ. CSQZPARM is in the APF-authorized library thlqual.SCSQAUTH also supplied with IBM MQ. The values of these parameters are displayed as a series of messages when you start IBM MQ.

See [START QMGR](#) for more information about how this command is used.

### Creating your own system parameter module

If CSQZPARM does not contain the system parameters you want, you can create your own system parameter module using the sample JCL provided in thlqual.SCSQPROC(CSQ4ZPRM).

To create your own system parameter module:

1. Make a working copy of the JCL sample.
2. Edit the parameters for each macro in the copy as required. If you remove any parameters from the macro calls, the default values are automatically picked up at run time.
3. Replace the placeholder ++NAME++ with the name that the load module is to take (this can be CSQZPARM).
4. If your assembler is not high-level assembler, change the JCL as required by your assembler.

5. Run the JCL to assemble and link edit the tailored versions of the system parameter macros to produce a load module. This is the new system parameter module with the name that you have specified.
6. Put the load module produced in an APF-authorized user library.
7. Add user READ access to the APF-authorized user library.
8. Include this library in the IBM MQ queue manager started task procedure STEPLIB. This library name must come before the library thlqual.SCSQAUTH in STEPLIB.
9. Invoke the new system parameter module when you start the queue manager. For example, if the new module is named NEWMODS, issue the command:

```
START QMGR PARM(NEWMODS)
```

10. Ensure successful completion of the command by checking the job log. There should be an entry in the log similar to the following:

```
CSQ9022I CDL1 CSQYASCP 'START QMGR' NORMAL COMPLETION
```

You can also specify the parameter module name in the queue manager startup JCL. For more information, see [Using MQSC to start and stop a queue manager on z/OS](#).

**Note:** If you choose to name your module CSQZPARM, you do not need to specify the PARM parameter on the START QMGR command.

## Fine tuning a system parameter module

IBM MQ also supplies a set of three assembler source modules, which can be used to fine-tune an existing system parameter module. These modules are in library thlqual.SCSQASMS. Typically, you use these modules in a test environment to change the default parameters in the system parameter macros. Each source module calls a different system parameter macro:

This assembler source module...	Calls this macro...
CSQFSYSP	CSQ6SYSP (connection and tracing parameters)
CSQJLOGP	CSQ6LOGP (log initialization)
CSQJARVP	CSQ6ARVP (archive initialization)

This is how you use these modules:

1. Make working copies of each assembler source module in a user assembler library.
2. Edit your copies by adding or altering the values of any parameters as required.
3. Assemble your copies of any edited modules to create object modules in a user object library.
4. Link edit these object code modules with an existing system parameter module to produce a load module that is the new system parameter module.
5. Ensure that new system parameter module is a member of a user authorized library.
6. Include this library in the queue manager started task procedure STEPLIB. This library must come before the library thlqual.SCSQAUTH in STEPLIB.
7. Invoke the new system parameter module by issuing a START QMGR command, specifying the new module name in the PARM parameter, as before.

A sample usermod is provided in member CSQ4UZPR of SCSQPROC which demonstrates how to manage customized system parameters under SMP/E control.

## Altering system parameters

You can alter some system parameters while a queue manager is running; see the [SET SYSTEM](#), [SET LOG](#), and [SET ARCHIVE](#) commands.

Put the SET commands in your initialization input data sets so that they take effect every time you start the queue manager.

### Related concepts

“Tailor the channel initiator parameters” on page 933

Use ALTER QMGR to customize the channel initiator to suit your requirements.

### Utilizzo di CSQ6SYSP

Utilizzare questo argomento come riferimento per l'impostazione dei parametri di sistema utilizzando CSQ6SYSP.

I parametri predefiniti per CSQ6SYSP, e se è possibile modificare ciascun parametro utilizzando il comando SET SYSTEM, sono riportati in Tabella 60 a pagina 912. Se si desidera modificare uno di questi valori, consultare le descrizioni dettagliate dei parametri.

Parametro	Descrizione	Valore predefinito	comando set
<a href="#">“ACCTIME” a pagina 914</a>	Il tempo, in minuti e secondi, tra ciascuna raccolta di dati di account.	-1	✓
<a href="#">“ACELIM” a pagina 914</a>	Dimensione del pool di archiviazione ACE in blocchi da 1 KB.	0 (nessun limite)	✓
<a href="#">“CACHE” a pagina 914</a>	Specifica il tipo di cache del cluster da utilizzare.	STATICO	-
<a href="#">“CMDUSER” a pagina 915</a>	L'ID utente predefinito per i controlli di sicurezza dei comandi.	CSQOPR	-
<a href="#">“EXCLMSG” a pagina 915</a>	Specifica un elenco di messaggi da escludere da qualsiasi registrazione. I messaggi in questo elenco non vengono inviati alla console z/OS e al log della copia cartacea. Di conseguenza, l'utilizzo del parametro EXCLMSG per escludere i messaggi è più efficiente da una prospettiva CPU rispetto all'utilizzo dei metodi descritti in <a href="#">“Suppress information messages” a pagina 938</a>	()	✓
<a href="#">“EXITLIM” a pagina 915</a>	Tempo massimo (in secondi) per cui le uscite del gestore code possono essere eseguite durante ogni chiamata.	30	-
<a href="#">“EXITTCB” a pagina 916</a>	Il numero di attività del server avviate da utilizzare per eseguire le uscite del gestore code.	8	-
<a href="#">“LOGLOAD” a pagina 916</a>	Numero di record di log scritti da IBM MQ tra l'avvio di un punto di controllo e il successivo.	500 000	✓
<a href="#">“MULCCAPT” a pagina 916</a>	Determina la proprietà Determinazione del prezzo dell'utilizzo misurato che controlla l'algoritmo per raccogliere i dati utilizzati da MULC (Measured Usage License Charging).	Vedere <a href="#">descrizione parametro</a>	-



Tabella 60. Valori predefiniti dei parametri CSQ6SYSP (Continua)

Parametro	Descrizione	Valore predefinito	comando set
<a href="#">“OTMACON” a pagina 916</a>	Parametri di connessione OTMA.	Vedere <a href="#">descrizione parametro</a>	-
<a href="#">“QINDXBLD” a pagina 917</a>	Determina se il riavvio del gestore code attende la ricreazione di tutti gli indici o viene completato prima della ricreazione di tutti gli indici.	ATTESA	-
<a href="#">“QMCCSID” a pagina 918</a>	CCSID (Coded character set identifier) per il gestore code.	Zero	-
<a href="#">“DATASGQ” a pagina 918</a>	Parametri del gruppo di condivisione code.	Vedere <a href="#">descrizione parametro</a>	-
<a href="#">“RESAUDIT” a pagina 919</a>	Parametro di controllo RESLEVEL.	Sì	-
<a href="#">“ROUTCDE” a pagina 919</a>	Il codice di instradamento del messaggio assegnato ai messaggi non sollecitati da una console specifica.	1	-
<a href="#">“SERVIZIO” a pagina 919</a>	Riservato per l'utilizzo da parte di IBM.	0	✓
<a href="#">“SMFACCT” a pagina 919</a>	Specifica se i dati di account SMF devono essere raccolti all'avvio del gestore code.  Tenere presente che i dati di account di canale di classe 4 vengono raccolti solo quando viene avviato l'inziatore di canali.	No	-
<a href="#">SMFSTAT</a>	Specifica se le statistiche SMF devono essere raccolte all'avvio del gestore code.  Tenere presente che i dati statistici dell'inziatore di canali di classe 4 vengono raccolti solo quando viene avviato l'inziatore di canali.	No	-
<a href="#">SPLCAP</a>	Specifica se la funzionalità della politica di sicurezza della coda è abilitata su questo gestore code. Per Advanced Message Security for z/OS, impostare questo parametro su YES.	No	-
<a href="#">STATIME</a>	Il tempo, in minuti e secondi, tra ogni raccolta di statistiche.	30	✓
<a href="#">TRACSTR</a>	Specifica se la traccia deve essere avviata automaticamente.	No	-
<a href="#">TRACTBL</a>	Dimensione della tabella di traccia, in blocchi da 4 KB, che deve essere utilizzata dalla funzione di traccia globale.	99 (396 KB)	✓
<a href="#">WLMTIME</a>	Intervallo di tempo tra la scansione dell'indice della coda per le code gestite da WLM.	30	-
<a href="#">WLMTIMU</a>	Unità (minuti o secondi) per WLMTIME.	minuti	-

## ACCTIME

Specifica l'intervallo, in minuti e secondi, tra le riunioni consecutive dei dati di account.

Specificare un numero, -1, o nell'intervallo compreso tra 0 e 1440 minuti nel formato 'mmm' oppure nell'intervallo tra 0 e 59 secondi, nel formato 'mm.ss'.

### Note:

- Quando si specifica solo un intervallo di secondi, è necessario anteporre all'intervallo il valore 0. L'intervallo più piccolo possibile è un secondo: '0.01'.
- Se si specifica un valore di 0, i dati di account vengono raccolti all'intervallo di registrazione globale SMF. Per ulteriori informazioni, consultare [Utilizzo di System Management Facility](#).
- Se si specifica un valore di -1, che è il valore predefinito, i dati di account vengono raccolti all'intervallo specificato dal valore STATIME.

Ad esempio:

'0.30' imposta un intervallo di 30 secondi.

'5.30' imposta un intervallo di 5 minuti e 30 secondi.

'30' imposta un intervallo di 30 minuti.

## ACELIM

Specifica la dimensione massima del pool di memoria ACE in blocchi da 1 KB. Il numero deve essere compreso tra 0 e 999999. Il valore predefinito zero indica nessun vincolo imposto, oltre a quanto è disponibile nel sistema.

È necessario impostare un valore solo per ACELIM sui gestori code che sono stati identificati come quelli che utilizzano quantità esorbitanti di memoria ECSA. Il limite del pool di memoria ACE ha l'effetto di limitare il numero di connessioni nel sistema, e quindi, la quantità di memoria ECSA utilizzata da un gestore code.

Una volta che il gestore code raggiunge il limite, non è possibile che le applicazioni ottengano nuove connessioni. La mancanza di nuove connessioni causa dei malfunzionamenti nell'elaborazione MQCONN e le applicazioni coordinate tramite RRS risconteranno probabilmente dei malfunzionamenti in qualsiasi API IBM MQ.

Un ACE rappresenta circa il 12,5% dell'ECSA totale richiesta per i blocchi di controllo correlati ai thread per una connessione. Quindi, ad esempio, specificando ACELIM=5120 si prevede di limitare la quantità totale di ECSA assegnata dal gestore code (per i blocchi di controllo relativi ai thread) a circa 40960K; , ossia 5120 moltiplicata per 8.

Per limitare la quantità totale di ECSA assegnata dal gestore code, per i blocchi di controllo correlati al thread a 5120K, è richiesto un valore ACELIM di 640.

È possibile utilizzare i record SMF 115 di sottotipo 5, prodotti dalla traccia CLASS(3) delle statistiche, per monitorare la dimensione del pool di memoria 'ACE/PEB' e impostare, quindi, un valore appropriato per ACELIM.

È possibile ottenere la quantità totale di memoria ECSA utilizzata dal gestore code, per i blocchi di controllo, dai record SMF 115 di sottotipo 7, scritti dalla traccia CLASS(2) delle statistiche. La quantità totale di memoria ECSA utilizzata è la somma dei campi QSRSPHBGF e QSRSPHBGV.

Per ulteriori informazioni sui record di statistiche SMF 115, consultare [Interpretazione delle statistiche delle prestazioni di IBM MQ](#).

Tenere presente che si dovrebbe considerare l'impostazione di ACELIM come meccanismo per proteggere un'immagine z/OS da un comportamento errato del gestore code, anziché come mezzo per controllare le connessioni dell'applicazione a un gestore code.

## CACHE

Specifica il tipo di cache del cluster da utilizzare.

La cache del cluster è un'area di memoria utilizzata per memorizzare le informazioni relative al cluster.

Se la cache del cluster è statica, ha una dimensione fissa, allocata all'avvio del gestore code. Se la cache si riempie, viene emesso il messaggio `CSQM060E` e la richiesta dell'applicazione che richiede più spazio riceve un `MQRC_CLUSTER_RESOURCE_ERROR`.

Se si imposta `CLCACHE` su dinamico, la cache del cluster può espandersi come richiesto. Tuttavia, è necessario prima assicurarsi che le uscite del carico di lavoro del cluster installate possano funzionare con una cache dinamica.

Se un'uscita del carico di lavoro del cluster installato non può funzionare con un messaggio della cache dinamica, viene emesso `CSQM061E`.

`MQXCLWLN` viene fornito per le uscite del carico di lavoro del cluster per navigare nella cache del cluster in un modo che funziona indipendentemente dal fatto che vengano utilizzate cache dinamiche o statiche.

Per i nuovi gestori code impostare `CLCACHE=DYNAMIC`, a meno che non si stia utilizzando un'uscita del carico di lavoro del cluster che non supporta la cache dinamica.

Per i gestori code esistenti che utilizzano già una cache statica e che si trovano in un cluster a cui non sono state aggiunte molte nuove code e gestori code, è ragionevole continuare a utilizzare `CLCACHE=STATIC`.

Per i gestori code esistenti che utilizzano già una cache statica e che si trovano in un cluster a cui verranno aggiunte molte nuove code o gestori code, iniziare a utilizzare `CLCACHE=DYNAMIC`.

### **STATICO**

Quando la cache del cluster è statica, la sua dimensione è fissa all'avvio del gestore code, sufficiente per la quantità corrente di informazioni sul cluster più spazio per l'espansione. La dimensione non può aumentare mentre il gestore code è attivo. Questa è l'opzione predefinita.

### **DINAMICO**

Quando la cache del cluster è dinamica, la dimensione iniziale assegnata all'avvio del gestore code può essere aumentata automaticamente se richiesto mentre il gestore code è attivo.

### **CMDUSER**

Specifica l'ID utente predefinito utilizzato per i controlli di sicurezza dei comandi. Questo ID utente deve essere definito in ESM (ad esempio, RACF). Specificare un nome composto da 1 a 8 caratteri alfanumerici. Il primo carattere deve essere un carattere alfabetico.

Il valore predefinito è `CSQOPR`.

### **EXCLMSG**

Specifica un elenco di messaggi di errore da escludere.

Questo elenco è dinamico e viene aggiornato utilizzando il comando `SET SYSTEM`.

Il valore predefinito è un elenco vuoto (`()`).

I messaggi vengono forniti senza il prefisso `CSQ` e senza il suffisso del codice azione (`I - D - E - A`). Ad esempio, per escludere il messaggio `CSQX500I`, aggiungere `X500` a questo elenco. Questo elenco può contenere un massimo di 16 identificativi di messaggio.

Per essere idoneo ad essere incluso nell'elenco, il messaggio deve essere emesso dopo il normale avvio degli spazi di indirizzo `MSTR` o `CHIN` e deve iniziare con uno dei seguenti caratteri `E, H, I, J, L, M, N, P, R, T, V, W, X, Y, 2, 3, 5, 9`.

Gli ID messaggio emessi come risultato dei comandi di elaborazione possono essere aggiunti all'elenco, tuttavia non verranno esclusi. Ad esempio, un identificativo di messaggio viene emesso come risultato del comando `DISPLAY USAGE PSID (*)`, tuttavia, questo messaggio non può essere eliminato.

### **EXITLIM**

Specifica il tempo, in secondi, consentito per ogni richiamo delle uscite del gestore code. (Questo parametro non ha alcun effetto sulle uscite del canale.)

Specificare un valore compreso tra 5 e 9999.

Il valore predefinito è 30. Il gestore code esegue il polling delle uscite in esecuzione ogni 30 secondi. Su ogni polling, tutti quelli che sono stati in esecuzione per un periodo di tempo superiore a quello specificato da EXITLIM vengono forzatamente terminati.

### **EXITTCB**

Specifica il numero di attività del server avviate da utilizzare per eseguire le uscite nel gestore code. (Questo parametro non ha alcun effetto sulle uscite del canale.) È necessario specificare un numero almeno pari al numero massimo di uscite (diverse dalle uscite del canale) che il gestore code potrebbe dover eseguire, altrimenti avrà esito negativo con una fine anomala 6c6 .

Specificare un valore compreso tra zero e 99. Un valore zero indica che non è possibile eseguire alcuna uscita.

Il valore predefinito è 8.

### **LOGLOAD**

Specifica il numero di record di log che IBM MQ scrive tra l'avvio di un checkpoint e il successivo. IBM MQ avvia un nuovo checkpoint dopo aver scritto il numero di record specificato.

Specificare un valore compreso tra 200 e 16 000 000.

Il valore predefinito è 500 000.

Maggiore è il valore, migliori sono le prestazioni di IBM MQ ; tuttavia, il riavvio richiede più tempo se il parametro è impostato su un valore elevato.

Impostazioni suggerite:

<b>Sistema di test</b>	10 000
<b>Sistema di produzione</b>	500 000

In un sistema di produzione, il valore predefinito fornito potrebbe risultare in una frequenza di checkpoint troppo elevata.

Il valore di LOGLOAD determina la frequenza dei checkpoint del gestore code. Un valore troppo grande significa che una grande quantità di dati viene scritta nel log tra i punti di controllo, determinando un aumento del tempo di riavvio del ripristino di inoltro del gestore code in seguito a un errore. Un valore troppo piccolo fa sì che i checkpoint si verifichino troppo frequentemente durante il carico di picco, influenzando negativamente i tempi di risposta e l'utilizzo del processore.

Per LOGLOAD si suggerisce un valore iniziale di 500 000. Per una frequenza di messaggi persistenti da 1 KB di 100 messaggi al secondo (ossia, 100 MQPUT con commit e 100 MQGET con commit) l'intervallo tra i punti di controllo è di circa 5 minuti.

**Nota:** Ciò è inteso solo come una linea guida e il valore ottimale per questo parametro dipende dalle caratteristiche del singolo sistema.

### **MULCCAPT**

Specifica l'algoritmo da utilizzare per la raccolta dei dati utilizzati da MULC (Measured Usage License Charging).

#### **STANDARD**

MULC è basato sull'ora dalla chiamata MQCONN API IBM MQ all'ora della chiamata MQDISC API IBM MQ .

#### **Dettagliato**

MULC si basa sul tempo che va dall'inizio di una chiamata API IBM MQ alla fine della chiamata API IBM MQ .

Il valore predefinito è STANDARD

### **OTMACON**

Parametri OTMA. Questa parola chiave accetta cinque parametri posizionali:

## **OTMACON = ( Group , Member , Druexit , Age , Tpipepfx)**

### **Gruppo**

Questo è il nome del gruppo XCF a cui appartiene questa particolare istanza di IBM MQ .

Può essere lungo da 1 a 8 caratteri e deve essere immesso in caratteri maiuscoli.

Il valore predefinito è vuoto, che indica che IBM MQ non deve tentare di unirsi a un gruppo XCF.

### **Membro**

Questo è il nome del membro di questa istanza particolare di IBM MQ all'interno del gruppo XCF.

Può contenere da 1 a 16 caratteri e deve essere immesso in caratteri maiuscoli.

Il valore predefinito è il nome del gestore code di 4 caratteri.

### **Druexit**

Specifica il nome dell'uscita utente di risoluzione della destinazione OTMA che deve essere eseguito da IMS.

Può contenere da 1 a 8 caratteri.

Il valore predefinito è DFSYDRU0.

Questo parametro è facoltativo; è obbligatorio se IBM MQ deve ricevere i messaggi da un'applicazione IMS che non è stata avviata da IBM MQ. Il nome deve corrispondere all'uscita utente di risoluzione di destinazione codificata nel sistema IMS . Per ulteriori informazioni, fare riferimento a [“Using OTMA exits in IMS”](#) a pagina 1001.

### **Età**

Rappresenta il periodo di tempo, in secondi, in cui un ID utente da IBM MQ viene considerato precedentemente verificato da IMS.

Può essere compreso tra zero e 2 147 483 647.

Il valore predefinito è 2 147 483 647.

Si consiglia di impostare questo parametro insieme al parametro `interval` del comando ALTER SECURITY per mantenere la coerenza delle impostazioni della cache di sicurezza nel mainframe.

### **Tpipepfx**

Rappresenta il prefisso da utilizzare per i nomi Tpipe.

Comprende tre caratteri; il primo carattere è compreso nell'intervallo da A a Z, i caratteri successivi sono da A a Z o da 0 a 9. Il valore predefinito è CSQ.

Viene utilizzato ogni volta che IBM MQ crea un Tpipe; il resto del nome viene assegnato da IBM MQ. Non è possibile impostare il nome Tpipe completo per qualsiasi Tpipe creato da IBM MQ.

## **QINDEXBLD**

Determina se il riavvio del gestore code attende la ricreazione di tutti gli indici della coda o viene completato prima della ricreazione di tutti gli indici.

### **ATTESA**

Il riavvio del gestore code attende il completamento di tutte le build dell'indice della coda. Ciò significa che non viene ritardata alcuna applicazione durante la normale elaborazione dell'API IBM MQ durante la creazione dell'indice, poiché tutti gli indici vengono creati prima che le applicazioni possano connettersi al gestore code.

Questa è l'opzione predefinita.

### **Nessuna attesa**

Il gestore code può essere riavviato prima che venga completata la creazione dell'indice della coda.

## QMCCSID

Specifica il CCSID (coded character set identifier) predefinito che il gestore code (e quindi l'accodamento distribuito) deve utilizzare.

Specificare un valore compreso tra zero e 65535. Il valore deve rappresentare una codepage EBCDIC elencata come codepage z/OS nativa per la lingua scelta in [Lingue nazionali](#).

Zero, che è il valore predefinito, significa utilizzare il CCSID correntemente impostato oppure, se non è impostato, utilizzare il CCSID 500. Ciò significa che se il CCSID è stato esplicitamente impostato su un valore diverso da zero, non è possibile reimpostarlo impostando QMCCSID su zero; è ora necessario utilizzare il CCSID diverso da zero corretto. Se QMCCSID è zero, è possibile controllare quale CCSID è attualmente in uso immettendo il comando DISPLAY QMGR CCSID.

**Nota:** Tutti i gestori code in un gruppo di condivisione code devono utilizzare lo stesso QMCCSID.

## DATASGQ

Dati del gruppo di condivisione code. Questa parola chiave accetta cinque parametri posizionali:

**QSGDATA = ( Qsgname , Dsgname , Db2name , Db2serv , Db2blob)**

### QSGNAME

Questo è il nome del gruppo di condivisione code a cui appartiene il gestore code.

Consultare [Regole per la denominazione degli oggetti IBM MQ](#) per i caratteri validi. Il nome:

- Può contenere da 1 a 4 caratteri
- Non deve iniziare con un valore numerico
- Non deve terminare con @.

Questo perché, per motivi di implementazione, i nomi con meno di quattro caratteri vengono riempiti internamente con simboli @,

Il valore predefinito è vuoto, che indica che il gestore code non è un membro di alcun gruppo di condivisione code.

### Nomegg

Questo è il nome del gruppo di condivisione dati Db2 a cui deve connettersi il gestore code.

Può essere lungo da 1 a 8 caratteri e deve essere immesso in caratteri maiuscoli.

Il valore predefinito è vuoto, che indica che non si stanno utilizzando gruppi di condivisione code.

### Db2name

Questo è il nome del sottosistema o del gruppo Db2 a cui deve connettersi il gestore code.

Può contenere da 1 a 4 caratteri e deve essere immesso in caratteri maiuscoli.

Il valore predefinito è vuoto, che indica che non si stanno utilizzando gruppi di condivisione code.

**Nota:** Il sottosistema Db2 (o l'allegato del gruppo) deve far parte del gruppo di condivisione dati Db2 specificato in Dsgnamee tutti i gestori code devono specificare lo stesso gruppo di condivisione dati Db2 .

### Db2serv

Questo è il numero di attività del server utilizzate per accedere a Db2.

Può essere compreso tra 4 e 10.

Il valore predefinito è 4.

### Db2blob

Questo è il numero di attività Db2 utilizzate per accedere ai BLOB (Binary Large Object).

Può essere compreso tra 4 e 10.

Il valore predefinito è 4.

Se si specifica uno dei parametri del nome (ovvero, **Qsgname**, **Dsgname** o **Db2name** ), è necessario immettere i valori per gli altri nomi, altrimenti IBM MQ non riesce.

#### **RESAUDIT**

Specifica se i record di controllo RACF vengono scritti per i controlli di sicurezza RESLEVEL eseguiti durante l'elaborazione della connessione.

Specificare uno tra:

**No**

Il controllo RESLEVEL non viene eseguito.

**sì**

Viene eseguito il controllo RESLEVEL.

Il valore predefinito è YES.

#### **ROUTCDE**

Specifica il codice di instradamento del messaggio z/OS predefinito assegnato ai messaggi che non vengono inviati in risposta diretta a un comando MQSC.

Specificare uno tra:

1. Un valore compreso nell'intervallo tra 1 e 16, inclusi.
2. Un elenco di valori, separati da una virgola e racchiusi tra parentesi. Ciascun valore deve essere compreso tra 1 e 16, inclusi.

Il valore predefinito è 1.

Per ulteriori informazioni sui codici di instradamento z/OS , consultare *Routing codes* in [Descrizione del messaggio](#) in uno dei volumi dei manuali *z/OS MVS System Messages* .

#### **SERVIZIO**

Questo campo è riservato per l'utilizzo da parte di IBM.

#### **SMFACCT**

Specifica se IBM MQ invia i dati di account a SMF automaticamente quando viene avviato il gestore code.

Specificare uno tra:

**No**

Non avviare automaticamente la raccolta dei dati di account.

**sì**

Avviare la raccolta automatica dei dati di account per la classe predefinita 1.

**numeri interi**

Un elenco di classi per cui i dati di account vengono raccolti automaticamente nell'intervallo compreso tra 1 e 4.

\* Avviare l'account SMF automaticamente per le classi 1, 2 e 3.

Il valore predefinito è NO.

#### **SMFSTAT**

Specifica se raccogliere le statistiche SMF automaticamente all'avvio del gestore code.

Specificare uno tra:

**No**

Non avviare automaticamente la raccolta delle statistiche.

**sì**

Avviare la raccolta automatica delle statistiche per la classe predefinita 1.

**numeri interi**

Un elenco di classi per cui le statistiche vengono raccolte automaticamente nell'intervallo compreso tra 1 e 5.

Per raccogliere le statistiche di classe 2 o 3, è necessario specificare anche la classe 1.

\* Avviare automaticamente le statistiche SMF per le classi 1, 2 e 3.

Il valore predefinito è NO.

### **SPLCAP**

La funzione della politica di sicurezza abilita un livello più elevato di sicurezza dei messaggi tramite politiche che controllano se i messaggi sono firmati o codificati, quando vengono scritti e letti dalle code.

L'elaborazione della politica di sicurezza è configurata per questo gestore code, impostando SPLCAP su uno dei seguenti valori:

#### **No**

La funzione di implementazione delle politiche di sicurezza dei messaggi per le code non viene abilitata durante l'inizializzazione del gestore code.

#### **Sì**

Le funzioni di sicurezza dei messaggi sono abilitate durante l'inizializzazione del gestore code.

Il gestore code controlla che l'attributo AMSPROD sia impostato su AMS, ADVANCED o ADVANCEDVUE, nel qual caso è stato concesso in licenza per AMS. Altrimenti non si avvierà.

Il gestore code controlla anche se è in atto la configurazione AMS necessaria. In caso contrario, il gestore code non verrà avviato.

Se il gestore code dispone della licenza per AMS e la configurazione necessaria è attiva, il gestore code verrà avviato con le funzionalità di sicurezza dei messaggi abilitate durante l'inizializzazione del gestore code e lo spazio di indirizzo AMSM viene avviato.

Il valore predefinito è NO.

### **STATIME**

Da IBM MQ for z/OS 9.3.0, specifica il tempo, in minuti e secondi, tra le raccolte consecutive di dati statistici. Se ACCTIME non è impostato, o è -1, specifica anche l'ora, tra le riunioni consecutive dei dati di account.

Specificare un numero compreso nell'intervallo tra 0 e 1440 minuti nel formato 'mmm' oppure nell'intervallo tra 0 e 1440 minuti e tra 0 e 59 secondi nel formato 'mmm.ss'. Il valore predefinito è 30 minuti.

#### **Note:**

- Quando si specifica solo un intervallo di secondi, è necessario anteporre all'intervallo il valore 0. L'intervallo più piccolo possibile è un secondo: '0.01'.
- Da IBM MQ for z/OS 9.3.0, se si specifica un valore di 0, i dati delle statistiche vengono raccolti al broadcast di raccolta dati SMF. Se ACCTIME non è specificato o è -1, i dati di account vengono raccolti anche al broadcast di raccolta dati SMF. Per ulteriori informazioni, consultare [Utilizzo di System Management Facility](#).
- Se si specifica un valore di -1, che è il valore predefinito, i dati di account vengono raccolti all'intervallo specificato dal valore STATIME.

### **TRACSTR**

Specifica se la traccia globale deve essere avviata automaticamente.

Specificare uno tra:

#### **No**

Non avviare automaticamente la traccia globale.

#### **Sì**

Avviare automaticamente la traccia globale per la classe predefinita, classe 1.

#### **numeri interi**

Un elenco di classi per cui la traccia globale deve essere avviata automaticamente nell'intervallo compreso tra 1 e 4.



\*

Avviare automaticamente la traccia globale per tutte le classi.

Il valore predefinito è NO se non si specifica la parola chiave nella macro.

**Nota:** Il modulo di caricamento del parametro di sistema predefinito fornito (CSQZPARM) ha TRACSTR=YES (impostato nel modulo assembler CSQFSYSP). Se non si desidera avviare la traccia automaticamente, creare il proprio modulo dei parametri di sistema o immettere il comando STOP TRACE dopo l'avvio del gestore code.

Per i dettagli sul comando STOP TRACE, consultare [STOP TRACE](#).

### TRACTBL

Specifica la dimensione predefinita, in blocchi da 4 KB, della tabella di traccia in cui la funzione di traccia globale memorizza i record di traccia IBM MQ .

Specificare un valore compreso tra 1 e 999.

Il valore predefinito è 99. Ciò equivale a 396 KB.

**Nota:** La memoria per la tabella di traccia è assegnata in ECSA. Pertanto, è necessario selezionare questo valore con attenzione.

### WLMTIME

Specifica il tempo (in minuti o secondi a seconda del valore di WLMTIMU) tra ogni scansione degli indici per le code gestite da WLM.

Specificare un valore compreso tra 1 e 9999.

Il valore predefinito è 30.

### WLMTIMU

Unità di tempo utilizzate con il parametro WLMTIME.

Specificare uno tra:

#### minuti

WLMTIME rappresenta un numero di minuti.

#### sec.

WLMTIME rappresenta un numero di secondi.

Il valore predefinito è MINS.

### Riferimenti correlati

[“Using CSQ6LOGP” a pagina 921](#)

Use this topic as a reference for how to specify logging options using CSQ6LOGP.

[“Using CSQ6ARVP” a pagina 926](#)

Use this topic as a reference for how to specify your archiving environment using CSQ6ARVP

### [Using CSQ6LOGP](#)


Use this topic as a reference for how to specify logging options using CSQ6LOGP.

Use CSQ6LOGP to establish your logging options.

The default parameters for CSQ6LOGP, and whether you can alter each parameter using the [SET LOG](#) command, are shown in [Default values of CSQ6LOGP parameters](#). If you need to change any of these values, refer to the detailed descriptions of the parameters.

Parameter	Description	Default value	SET command
<a href="#">COMPLOG</a>	Controls whether log compression is enabled.	NONE	X

Table 61. Default values of CSQ6LOGP parameters (continued)

Parameter	Description	Default value	SET command
<u>DEALLCT</u>	Length of time an archive tape unit remains unused before it is deallocated.	zero	X
<u>INBUFF</u>	Size of input buffer storage for active and archive log data sets.	60 KB	-
<u>MAXARCH</u>	Maximum number of archive log volumes that can be recorded.	500	X
<u>MAXCNOFF</u>	Maximum number of CSQJOFF7 offload tasks that can be run in parallel.	31	-
<u>MAXRTU</u>	Maximum number of dedicated tape units allocated to read archive log tape volumes concurrently.	2	X
<u>OFFLOAD</u>	Archiving on or off.	YES (ON)	-
<u>OUTBUFF</u>	Size of output buffer storage for active and archive log data sets.	4 000 KB	-
<u>TWOACTV</u>	Single or dual active logging.	YES (dual)	-
<u>TWOARCH</u>	Single or dual archive logging.	YES (dual)	-
<u>TWOBSDS</u>	Single or dual BSDS.	YES (dual BSDS)	-
<u>WRTHRSH</u>	Number of output buffers to be filled before they are written to the active log data sets.	20	X
<u>ZHYWRITE</u>	Specifies whether the zHyperWrite feature is enabled.	NO	X
 <u>ZHYLINK</u>	Specifies whether the zHyperLink feature is enabled.	NO	X

### COMPLOG

Specifies whether log compression is enabled.

Specify either:

#### NONE

Log compression is not enabled.

#### RLE

Log compression is enabled using run-length encoding.

#### ANY

The queue manager selects the compression algorithm that gives the greatest degree of log record compression. This option results in RLE compression.

The default is NONE.

For more details about log compression, see [Log compression](#).

### DEALLCT

Specifies the length of time, in minutes, that an archive read tape unit is allowed to remain unused before it is deallocated.

Specify one of the following:

- Time, in minutes, in the range zero through 1440
- NOLIMIT

Specifying 1440 or NOLIMIT means that the tape unit is never deallocated.

The default is zero.

When archive log data is being read from tape, it is recommended that you set this value high enough to allow IBM MQ to optimize tape handling for multiple read applications.

### **INBUFF**

Specifies the size, in kilobytes, of the input buffer for reading the active and archive logs during recovery. Use a decimal number in the range 28 through 60. The value specified is rounded up to a multiple of 4.

The default is 60 KB.

Suggested settings:

**Test system** 28 KB

**Production system** 60 KB

Set this to the maximum for best log read performance.

### **MAXARCH**

Specifies the maximum number of archive log volumes that can be recorded in the BSDS. When this number is exceeded, recording begins again at the start of the BSDS.

Use a decimal number in the range 10 through 1000.

The default is 500.

Suggested settings:

**Test system** 500 (default)

**Production system** 1 000

Set this to the maximum so that the BSDS can record as many logs as possible.

For information about the logs and BSDS, see [Managing IBM MQ resources](#).

### **MAXCNOFF**

Specifies the number of CSQJOFF7 offload tasks that can be run in parallel.

This allows a queue manager, or queue managers, to be tuned such that they will not use all the available tape units.

Instead the queue manager waits until a CSQJOFF7 offload task has completed before trying to allocate any new archive data sets.

If the queue manager is archiving to tape, set this parameter so that the number of concurrent tape requests should not equal, or exceed, the number of tape units available, otherwise the system might hang.

Note that if dual archiving is in use, then each offload task performs both archives, so the parameter needs to be set accordingly. For example if the queue manager is dual archiving to tape, a value of MAXCNOFF=2 would allow up to two active logs to be archived concurrently to four tapes.

If several queue managers are sharing the tape units, you should set the MAXCNOFF for each queue manager accordingly.

The default value is 31.

Specify a value in the range 1 through 31.

### **MAXRTU**

Specifies the maximum number of dedicated tape units that can be allocated to read archive log tape volumes concurrently.

This parameter and the DEALLCT parameter allow IBM MQ to optimize archive log reading from tape devices.

Specify a value in the range 1 through 99.

The default is 2.

It is recommended that you set the value to be at least one less than the number of tape units available to IBM MQ. If you do otherwise, the offload process could be delayed, which could affect the performance of your system. For maximum throughput during archive log processing, specify the largest value possible for this option, remembering that you need at least one tape unit for offload processing.

#### **OFFLOAD**

Specifies whether archiving is on or off.

Specify either:

##### **YES**

Archiving is on

##### **NO**

Archiving is off

The default is YES.

**Attention:** Do **not** switch archiving off unless you are working in a test environment. If you do switch it off, you cannot guarantee that data will be recovered in the event of a system or transaction failure.

#### **OUTBUFF**

Specifies the total size, in kilobytes, of the storage to be used by IBM MQ for output buffers for writing the active and archive log data sets. Each output buffer is 4 KB.

The parameter must be in the range 128 through 4000. The value specified is rounded up to a multiple of 4. Values between 40 and 128 will be accepted for compatibility reasons, and are treated as a value of 128.

The default is 4000 KB.

Suggested settings:

<b>Test system</b>	400 KB
<b>Production system</b>	4 000 KB

Set this value to the maximum to avoid running out of log output buffers.

#### **TWOACTV**

Specifies single or dual active logging.

Specify either:

##### **NO**

Single active logs

##### **YES**

Dual active logs

The default is YES.

For more information about the use of single and dual logging, see [Managing IBM MQ resources](#).

#### **TWOARCH**

Specifies the number of archive logs that IBM MQ produces when the active log is offloaded.

Specify either:

##### **NO**

Single archive logs

**YES**

Dual archive logs

The default is YES.

Suggested settings:

**Test system** NO

**Production system** YES (default)

For more information about the use of single and dual logging, see [Managing IBM MQ resources](#).

**TWOBSDS**

Specifies the number of bootstrap data sets.

Specify either:

**NO**

Single BSDS

**YES**

Dual BSDS

The default is YES.

For more information about the use of single and dual logging, see [Managing IBM MQ resources](#).

**WRTHRSR**

Specifies the number of 4 KB output buffers to be filled before they are written to the active log data sets.

The larger the number of buffers, the less often the write takes place, and this improves the performance of IBM MQ. The buffers might be written before this number is reached if significant events, such as a commit point, occur.

Specify the number of buffers in the range 1 through 256.

The default is 20.

**ZHYWRITE**

Specifica se le scritture sui log attivi vengono effettuate con zHyperWrite abilitato.

Per ulteriori informazioni, vedi [Utilizzo di zHyperWrite with IBM MQ active logs](#).

Il valore può essere:

**No**

zHyperWrite non è abilitato.

**sì**

zHyperWrite è abilitato.

**V 9.4.0 ZHYLINK**

Specifies whether writes to the active logs are made with zHyperLink being enabled.

For more information on enabling active logs with zHyperLink, see [Using zHyperLink with IBM MQ](#).

The value can be:

**NO**

zHyperLink is not enabled.

**YES**

zHyperLink is enabled.

**Note:** Enabling ZHYLINK also enables ZHYWRITE

**Related reference**

[“Utilizzo di CSQ6SYSP” on page 912](#)

Utilizzare questo argomento come riferimento per l'impostazione dei parametri di sistema utilizzando CSQ6SYSP.

[“Using CSQ6ARVP” on page 926](#)

Use this topic as a reference for how to specify your archiving environment using CSQ6ARVP

**z/OS** *Using CSQ6ARVP*

Use this topic as a reference for how to specify your archiving environment using CSQ6ARVP

Use CSQ6ARVP to establish your archiving environment.

The default parameters for CSQ6ARVP, and whether you can alter each parameter using the SET ARCHIVE command, are shown in Table 62 on page 926. If you need to change any of these values, refer to the detailed descriptions of the parameters. For more information about planning your storage, see [Planning your storage and performance requirements on z/OS](#) .

Parameter	Description	Default value	SET command
<a href="#">ALCUNIT</a>	Units in which primary and secondary space allocations are made.	BLK (blocks)	X
<a href="#">ARCPFX1</a>	Prefix for first archive log data set name.	CSQARC1	X
<a href="#">ARCPFX2</a>	Prefix for second archive log data set name.	CSQARC2	X
<a href="#">ARCRETN</a>	The retention period of the archive log data set in days.	9999	X
<a href="#">ARCWRTC</a>	List of route codes for messages to the operator about archive log data sets.	1,3,4	X
<a href="#">ARCWTOR</a>	Whether to send message to operator and wait for reply before trying to mount an archive log data set.	YES	X
<a href="#">BLKSIZE</a>	Block size of archive log data set.	28 672	X
<a href="#">CATALOG</a>	Whether archive log data sets are cataloged in the ICF.	NO	X
<a href="#">COMPACT</a>	Whether archive log data sets should be compacted.	NO	X
<a href="#">PRIQTY</a>	Primary space allocation for DASD data sets.	25 715	X
<a href="#">PROTECT</a>	Whether archive log data sets are protected by ESM profiles when the data sets are created.	NO	X
<a href="#">QUIESCE</a>	Maximum time, in seconds, allowed for quiesce when ARCHIVE LOG with MODE(QUIESCE) specified.	5	X
<a href="#">SECQTY</a>	Secondary space allocation for DASD data sets. See the ALCUNIT parameter for the units to be used.	540	X
<a href="#">TSTAMP</a>	Whether the archive data set name should include a time stamp.	NO	X
<a href="#">UNIT</a>	Device type or unit name on which the first copy of archive log data sets is stored.	TAPE	X
<a href="#">UNIT2</a>	Device type or unit name on which the second copy of archive log data sets is stored.	Blank	X

**ALCUNIT**

Specifies the unit in which primary and secondary space allocations are made.

Specify one of:

**CYL**

Cylinders

**TRK**

Tracks

**BLK**

Blocks

You are recommended to use BLK because it is independent of the device type.

The default is BLK.

If free space on the archive DASD volumes is likely to be fragmented, you are recommended to specify a smaller primary extent and allow expansion into secondary extents. For more information about space allocation for active logs, refer to [Planning your log archive storage](#).

**ARCPFX1**

Specifies the prefix for the first archive log data set name.

See the TSTAMP parameter for a description of how the data sets are named and for restrictions on the length of ARCPFX1.

This parameter cannot be left blank.

The default is CSQARC1.

You might need to authorize the userid associated with the IBM MQ queue manager address space to create archive logs with this prefix.

**ARCPFX2**

Specifies the prefix for the second archive log data set name.

See the TSTAMP parameter for a description of how the data sets are named and for restrictions on the length of ARCPFX2.

This parameter cannot be blank even if the TWOARCH parameter is specified as NO.

The default is CSQARC2.

You might need to authorize the userid associated with the IBM MQ queue manager address space to create archive logs with this prefix.

**ARCRETN**

Specifies the retention period, in days, to be used when the archive log data set is created.

The parameter must be in the range zero through 9999.

The default is 9999.

Suggested settings:

**Test system** 3

In a test system, archive logs are probably not required over long periods.

**Production system** 9 999 (default)

Set this value high to effectively switch automatic archive log deletion off.

For more information about discarding archive log data sets, see [Discarding archive log data sets](#).

**ARCWRTC**

Specifies the list of z/OS routing codes for messages about the archive log data sets to the operator. This field is ignored if ARCWTOR is set to NO.

Specify up to 14 routing codes, each with a value in the range 1 through 16. You must specify at least one code. Separate codes in the list by commas, not by blanks.

The default is the list of values: 1,3,4.

For more information about z/OS routing codes, see *Routing codes* in *Message description* in one of the volumes of the *z/OS MVS System Messages* manuals.

### **ARCWTOR**

Specifies whether a message is to be sent to the operator and a reply is received before attempting to mount an archive log data set.

Other IBM MQ users might be forced to wait until the data set is mounted, but they are not affected while IBM MQ is waiting for the reply to the message.

Specify either:

#### **YES**

The device needs a long time to mount archive log data sets. For example, a tape drive.

#### **NO**

The device does not have long delays. For example, DASD.

The default is YES.

Suggested settings:

**Test system** NO

**Production system** YES (default)

This is dependent on operational procedures. If tape robots are used, NO might be more appropriate.

### **BLKSIZE**

Specifies the block size of the archive log data set. The block size you specify must be compatible with the device type you specify in the UNIT parameter.

The parameter must be in the range 4 097 through 28 672. The value you specify is rounded up to a multiple of 4 096.

The default is 28 672.

This parameter is overridden by the storage management subsystem (SMS) data class blocksize, if it is provided.

If the archive log data set is written to DASD, you are recommended to choose the maximum block size that allows two blocks for each track. For example, for a 3390 device, you should use a block size of 24 576.

If the archive log data set is written to tape, specifying the largest possible block size improves the speed of reading the archive log. You should use a block size of 28 672.

Suggested settings:

**Test system** Use the block size recommendation depending on the media used for archive logs.

That is, for disk 24 576, and tape 28 672.

**Production system** Use the block size recommendation depending on the media used for archive logs.

That is, for disk 24 576, and tape 28 672.

### **CATALOG**

Specifies whether archive log data sets are cataloged in the primary integrated catalog facility (ICF) catalog.



Specify either:

**NO**

Archive log data sets are not cataloged

**YES**

Archive log data sets are cataloged

The default is NO.

All archive log data sets allocated on DASD must be cataloged. If you archive to DASD with the CATALOG parameter set to NO, message [CSQJ072E](#) is displayed each time an archive log data set is allocated, and IBM MQ catalogs the data set.

Suggested settings:

**Test system** YES

**Production system** YES, when archives are allocated on DASD

**COMPACT**

Specifies whether data written to archive logs is to be compacted. This option applies only to a 3480 or 3490 device that has the improved data recording capability (IDRC) feature. When this feature is turned on, hardware in the tape control unit writes data at a much higher density than normal, allowing for more data on each volume. Specify NO if you do not use a 3480 device with the IDRC feature or a 3490 base model, except for the 3490E. Specify YES if you want the data to be compacted.

Specify either:

**NO**

Do not compact the data sets

**YES**

Compact the data sets

The default is NO.

Specifying YES adversely affects performance. Also be aware that data compressed to tape can be read only using a device that supports the IDRC feature. This can be a concern if you have to send archive tapes to another site for remote recovery.

Suggested settings:

**Test system** Not applicable

**Production system** NO (default)

This applies to 3480 and 3490 IDR compression only. Setting this to YES might degrade archive log read performance during recovery and restart; however, it does not affect writing to tape.

**PRIQTY**

Specifies the primary space allocation for DASD data sets in ALCUNITs.

The value must be greater than zero.

The default is 25 715.

This value must be sufficient for a copy of either the log data set or its corresponding BSDS, whichever is the larger. To determine the necessary value, follow this procedure:

1. Determine the number of active log records allocated ( c ) as explained in [“Create the bootstrap and log data sets”](#) on page 908.
2. Determine the number of 4096 byte blocks in each archive log block:

$$d = \text{BLKSIZE} / 4096$$

where BLKSIZE is the rounded up value.

### 3. If ALCUNIT=BLK:

$$\text{PRIQTY} = \text{INT}(c / d) + 1$$

where INT means round down to an integer.

### If ALCUNIT=TRK:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}))) + 1$$

where e is the number of bytes for each track (56664 for a 3390 device) and INT means round down to an integer.

### If ALCUNIT=CYL:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}) * f)) + 1$$

where f is the number of tracks for each cylinder (15 for a 3390 device) and INT means round down to an integer.

For information about how large to make your log and archive data sets, see [“Create the bootstrap and log data sets”](#) on page 908 and [“Define your page sets”](#) on page 909.

Suggested settings:

**Test system** 1 680

Sufficient to hold the entire active log, that is:

$$10\ 080 / 6 = 1\ 680 \text{ blocks}$$

**Production system** Not applicable when archiving to tape.

If free space on the archive DASD volumes is likely to be fragmented, you are recommended to specify a smaller primary extent and allow expansion into secondary extents. For more information about space allocation for active logs, see [Planning your log archive storage](#).

## PROTECT

Specifies whether archive log data sets are to be protected by discrete ESM (external security manager) profiles when the data sets are created.

Specify either:

### NO

Profiles are not created.

### YES

Discrete data set profiles are created when logs are offloaded. If you specify YES:

- ESM protection must be active for IBM MQ.

- The user ID associated with the IBM MQ queue manager address space must have authority to create these profiles.
- The TAPEVOL class must be active if you are archiving to tape.

Otherwise, offloading fails.

The default is NO.

### **QUIESCE**

Specifies the maximum time in seconds allowed for the quiesce when an ARCHIVE LOG command is issued with MODE(QUIESCE) specified.

The parameter must be in the range 1 through 999.

The default is 5.

### **SECQTY**

Specifies the secondary space allocation for DASD data sets in ALCUNITs. The secondary extent can be allocated up to 15 times; see the [IBM z/OS Management Facility Programming Guide](#) for more information on ALCUNIT.

The parameter must be greater than zero.

The default is 540.

### **TSTAMP**

Specifies whether the archive log data set name has a time stamp in it.

Specify either:

#### **NO**

Names do not include a time stamp. The archive log data sets are named:

```
arcpxi.A nnnnnn
```

Where *arcpxi* is the data set name prefix specified by ARCPFX1 or ARCPFX2. *arcpxi* can have up to 35 characters.

#### **YES**

Names include a time stamp. The archive log data sets are named:

```
arcpxi.cydd.T hhmsst.A nnnnnn
```

where *c* is 'D' for the years up to and including 1999 or 'E' for the year 2000 and later, and *arcpxi* is the data set name prefix specified by ARCPFX1 or ARCPFX2. *arcpxi* can have up to 19 characters.

#### **EXT**

Names include a time stamp. The archive log data sets are named:

```
arcpxi.D yyyydd.T hhmsst.A nnnnnn
```

Where *arcpxi* is the data set name prefix specified by ARCPFX1 or ARCPFX2. *arcpxi* can have up to 17 characters.

The default is NO.

### **UNIT**

Specifies the device type or unit name of the device that is used to store the first copy of the archive log data set.

Specify a device type or unit name of 1 through 8 alphanumeric characters. The first character must be alphabetic.

This parameter cannot be blank.

The default is TAPE.

If you archive to DASD, you can specify a generic device type with a limited volume range, for example, UNIT=3390.

If you archive to DASD, make sure that:

- The primary space allocation is large enough to contain all the data from the active log data sets.
- The archive log data set catalog option (CATALOG) is set to YES.
- You have used a proper value for BLKSIZE.

If you archive to TAPE, IBM MQ can extend to a maximum of 20 volumes.

Suggested settings:

<b>Test system</b>	DASD
<b>Production system</b>	TAPE

For more information about choosing a location for archive logs, see [Planning your log archive storage](#).

## UNIT2

Specifies the device type or unit name of the device that is used to store the second copy of the archive log data sets.

Specify a device type or unit name of 1 through 8 alphanumeric characters. The first character must be alphabetic. If this parameter is blank, the value set for the UNIT parameter is used.

The default is blank.

## Related reference

[“Utilizzo di CSQ6SYSP” on page 912](#)

Utilizzare questo argomento come riferimento per l'impostazione dei parametri di sistema utilizzando CSQ6SYSP.

[“Using CSQ6LOGP” on page 921](#)

Use this topic as a reference for how to specify logging options using CSQ6LOGP.

 [Using CSQ6USGP](#)

Use this topic as a reference for how to set your system parameters using CSQ6USGP

Use CSQ6USGP to control product usage recording.

The default parameters for CSQ6USGP are shown in [Table 63 on page 932](#). If you need to change any of these values, refer to the detailed descriptions of the parameters.



**Attention:** You cannot alter any of these parameters using the SET SYSTEM command.

Parameter	Description	Default value
<a href="#">QMGRPROD</a>	Product against which queue manager usage is to be recorded	Blank
<a href="#">AMSPROD</a>	Product against which Advanced Message Security (AMS) usage is to be recorded	Blank

## QMGRPROD

Specifies the product against which queue manager usage is to be recorded.

Specify one of:

### MQ

Queue manager usage is recorded as a stand-alone IBM MQ for z/OS product, with product ID 5655-MQ9.

**VUE**

Queue manager usage is recorded as a stand-alone IBM MQ for z/OS Value Unit Edition (VUE) product, with product ID 5655-VU9.

**ADVANCEDVUE**

Queue manager usage is recorded as part of an IBM MQ Advanced for z/OS Value Unit Edition product, with product ID 5655-AV1.

**AMSPROD**

If this parameter is not set the AMS address space will not start up and message [CSQY024I](#) will be output.

Specifies the product against which Advanced Message Security usage is to be recorded, if used.

Specify one of:

**AMS**

AMS usage is recorded as a stand-alone Advanced Message Security for z/OS product, with product ID 5655-AM9.

**ADVANCED**

AMS usage is recorded as part of an IBM MQ Advanced for z/OS product, with product ID 5655-AV9.

**ADVANCEDVUE**

AMS usage is recorded as part of an IBM MQ Advanced for z/OS Value Unit Edition product, with product ID 5655-AV1.

See [Reporting product information](#) for more information on product usage recording.

**Related reference**

[“Utilizzo di CSQ6SYSP” on page 912](#)

Utilizzare questo argomento come riferimento per l'impostazione dei parametri di sistema utilizzando CSQ6SYSP.

[“Using CSQ6LOGP” on page 921](#)

Use this topic as a reference for how to specify logging options using CSQ6LOGP.

 **Tailor the channel initiator parameters**

Use ALTER QMGR to customize the channel initiator to suit your requirements.

- *Repeat this task for each IBM MQ queue manager, as required.*
- *You must perform this task when migrating from a previous version.*

A number of queue manager attributes control how distributed queuing operates. Set these attributes using the MQSC command ALTER QMGR. The initialization data set sample thlqual.SCSQPROC(CSQ4INYG) contains some settings that you can customize. For more information, see [ALTER QMGR](#).

The values of these parameters are displayed as a series of messages each time you start the channel initiator.

**The relationship between adapters, dispatchers, and maximum number of channels**

The ALTER QMGR parameters CHIADAPS and CHIDISPS define the number of task control blocks (TCBs) used by the channel initiator. CHIADAPS (adapter) TCBs are used to make IBM MQ API calls to the queue manager. CHIDISPS (dispatcher) TCBs are used to make calls to the communications network.

The ALTER QMGR parameter MAXCHL influences the distribution of channels over the dispatcher TCBs.

**CHIDISPS**

If you have a small number of channels use the default value.

One task for each processor optimizes system performance. As dispatcher tasks are CPU intensive, the principle is to keep as few tasks as busy as possible, so that the time taken to find and start threads is minimized.

CHDISPS(20) is suitable for systems with more than 100 channels. There is unlikely to be any significant disadvantage in having CHDISPS(20) where this is more dispatcher TCBs than necessary.

As a guideline, if you have more than 1000 channels, allow one dispatcher for every 50 current channels. For example, specify CHDISPS(40) to handle up to 2000 active channels.

If you are using TCP/IP, the maximum number of dispatchers used for TCP/IP channels is 100, even if you specify a larger value in CHDISPS.

### **CHIADAPS**

Each IBM MQ API call to the queue manager is independent of any other and can be made on any adapter TCB. Calls using persistent messages can take much longer than those for nonpersistent messages because of log I/O. Thus a channel initiator processing a large number of persistent messages across many channels may need more than the default 8 adapter TCBs for optimum performance. This is particularly so where achieved batchsize is small, because end of batch processing also requires log I/O, and where thin client channels are used.

The suggested value for a production environment is CHIADAPS(30). Using more than this is unlikely to give any significant extra benefit, and there is unlikely to be any significant disadvantage in having CHIADAPS(30) if this is more adapter TCBs than necessary.

### **MAXCHL**

Each channel is associated with a particular dispatcher TCB at channel start and remains associated with that TCB until the channel stops. Many channels can share each TCB. MAXCHL is used to spread channels across the available dispatcher TCBs. The first (  $\text{MIN}(\text{MAXCHL} / \text{CHDISPS}), 10$  ) channels to start are associated with the first dispatcher TCB, and so on, until all dispatcher TCBs are in use.

The effect of this for small numbers of channels and a large MAXCHL is that channels are NOT evenly distributed across dispatchers. For example, if you set CHDISPS(10) and left MAXCHL at its default value of 200 but had only 50 channels, five dispatchers would be associated with 10 channels each and five would be unused. We suggest setting MAXCHL to the number of channels actually to be used where this is a small fixed number.

If you change this queue manager property, you must also review the ACTCHL, LU62CHL, and TCPCHL queue manager properties to ensure that the values are compatible. See [Queue manager parameters](#) for a full description of these properties, and their relationship.

## **Setting up your z/OS UNIX System Services environment for channel initiators**

The channel initiator (CHINIT) uses OMVS threads. Review the OMVS configuration parameters before creating a new CHINIT, or modifying the number of dispatchers or SSLTASKS.

Each CHINIT uses  $3 + \text{CHDISP} + \text{SSLTASKS}$  OMVS threads. These contribute to the total number of OMVS threads used in the LPAR, and towards the number of threads used by CHINIT started task user ID.

You can use the **D OMVS, L** and review the current usage, highwater usage, and system limit of MAXPROCSYS (the maximum number of processes that the system allows).

If you are adding a new CHINIT or increasing the values of CHDISPS or SSLTASKS then you must calculate the increase in threads and review the impact on the MAXPROCSYS values. You can use the **SETOMVS** command to dynamically change the MAXPROCSYS, or update the BPXPRCxx parmlib value or both.

The OMVS parameter MAXPROCUSER is the number of OMVS threads a single OMVS user, that is with the same UID, can have. The threads count towards this value. So if you have 2 CHINITs with the same started task user ID, with 10 dispatchers and 3 SSLTASKS each then there are  $2 * (3 + 10 + 3) = 32$  threads for the OMVS uid.

You can display the default MAXPROCUSER by issuing the **D OMVS,0** command and you can use the **SETOMVS** command to dynamically change the MAXPROCUSER, or update the BPXPRCxx parmlib value or both.

You can override this value on a per user basis with the RACF command **ALTUSER userid OMVS (PROCUSERMAX(nnnn))** or equivalent.

To start the channel initiator, issue the following command:

```
START CHINIT
```

To ensure that the channel initiator has started successfully, check that there is no ICH408I error in the xxxxCHIN(ssidCHIN) job log.

### Related concepts

[“Set up Batch, TSO, and RRS adapters” on page 935](#)

Make the adapters available to applications by adding libraries to appropriate STEPLIB concatenations. To cater for SNAP dumps issued by an adapter, allocate a CSQSNAP DDname. Consider using CSQBDEFV to improve the portability of your application programs

### Related reference

[Channel initiator statistics data records](#)

## **Set up Batch, TSO, and RRS adapters**

Make the adapters available to applications by adding libraries to appropriate STEPLIB concatenations. To cater for SNAP dumps issued by an adapter, allocate a CSQSNAP DDname. Consider using CSQBDEFV to improve the portability of your application programs

- *Repeat this task for each IBM MQ queue manager as required.*
- *You might need to perform this task when migrating from a previous version.*

To make the adapters available to batch and other applications using batch connections, add the following IBM MQ libraries to the STEPLIB concatenation for your batch application :

- thlqual.SCSQANL x
- thlqual.SCSQAUTH

where x is the language letter for your national language. (You do not need to do this if the libraries are in the LPA or the link list.)

For TSO applications add the libraries to the STEPLIB concatenation in the TSO logon procedure or activate them using the TSO command TSOLIB.

If the adapter detects an unexpected IBM MQ error, it issues an z/OS SNAP dump to DDname CSQSNAP, and issues reason code MQRC\_UNEXPECTED\_ERROR to the application. If the CSQSNAP DD statement is not in the application JCL or CSQSNAP is not allocated to a data set under TSO, no dump is taken. If this happens, you could include the CSQSNAP DD statement in the application JCL or allocate CSQSNAP to a data set under TSO and rerun the application. However, because some problems are intermittent, it is recommended that you include a CSQSNAP statement in the application JCL or allocate CSQSNAP to a data set in the TSO logon procedure to capture the reason for failure at the time it occurs.

The supplied program CSQBDEFV improves the portability of your application programs. In CSQBDEFV, you can specify the name of a queue manager, or queue sharing group, to be connected to rather than specifying it in the MQCONN or MQCONNX call in an application program. You can create a new version of CSQBDEFV for each queue manager, or queue sharing group. To do this, follow these steps:

1. Copy the IBM MQ assembler program CSQBDEFV from thlqual.SCSQASMS to a user library.
2. The supplied program contains the default subsystem name CSQ1. You can retain this name for testing and installation verification. For production subsystems, you can change the NAME=CSQ1 to your one- to four-character subsystem name, or use CSQ1.

If you are using queue sharing groups, you can specify a queue sharing group name instead of CSQ1. If you do this, the program issues a connect request to an active queue manager within that group.

3. Assemble and link-edit the program to produce the CSQBDEFV load module. For the assembly, include the library thlqual.SCSQMACS in your SYSLIB concatenation; use the link-edit parameters RENT, AMODE=31, RMODE=ANY. This is shown in the sample JCL in thlqual.SCSQPROC(CSQ4DEFV). Then include the load library in the z/OS Batch or the TSO STEPLIB, ahead of thlqual.SCSQAUTH.

### Related concepts

[“Set up the operations and control panels” on page 936](#)

To set up the operations and control panels you must first set up the libraries that contain the required panels, EXECs, messages, and tables. To do this, you must take into account which national language feature is to be used for the panels. When you have done this, you can optionally update the main ISPF menu for IBM MQ operations and control panels and change the function key settings.

### **Set up the operations and control panels**

To set up the operations and control panels you must first set up the libraries that contain the required panels, EXECs, messages, and tables. To do this, you must take into account which national language feature is to be used for the panels. When you have done this, you can optionally update the main ISPF menu for IBM MQ operations and control panels and change the function key settings.

- *You need to perform this task once for each z/OS system where you want to run IBM MQ.*
- *You might need to perform this task when migrating from a previous version.*

## Setting up the libraries

Follow these steps to set up the IBM MQ operations and control panels:

1. Ensure that all the libraries contained in your concatenations are either in the same format (F, FB, V, VB) and have the same block size, or are in order of decreasing block sizes. Otherwise, you might have problems trying to use these panels.
2. Include the library thlqual.SCSQEXEC in your SYSEXEC or SYSPROC concatenation or activate it using the TSO ALTLIB command. This library, which is allocated with a fixed-block 80 record format during installation, contains the required EXECs.

It is preferable to put the library into your SYSEXEC concatenation. However, if you want to put it in SYSPROC, the library must have a record length of 80 bytes.

3. Add thlqual.SCSQAUTH and thlqual.SCSQANLx to the TSO logon procedure STEPLIB or activate it using the TSO TSOLIB command, if it is not in the link list or the LPA.
4. You can either add the IBM MQ panel libraries permanently to your ISPF library setup, or allow them to be set up dynamically when the panels are used. For the former choice, you need to do the following:
  - a. Include the library containing the operations and control panel definitions in your ISPLLIB concatenation. The name is thlqual.SCSQPNLx, where x is the language letter for your national language.
  - b. Include the library containing the required tables in your ISPTLIB concatenation. The name is thlqual.SCSQTBLx, where x is the language letter for your national language.
  - c. Include the library containing the required messages in your ISPLMLIB concatenation. The name is thlqual.SCSQMSGx, where x is the language letter for your national language.
  - d. Include the library containing the required load modules in your ISPLLIB concatenation. The name of this library is thlqual.SCSQAUTH.

For the latter choice, use the z/OS [LIBDEF](#) command. See [Examples](#) for a link to various keywords you can use.

5. Test that you can access the IBM MQ panels from the TSO Command Processor panel. This is usually option 6 on the ISPF/PDF Primary Options Menu. The name of the EXEC that you run is CSQOREXX. There are no parameters to specify if you have put the IBM MQ libraries permanently in your ISPF setup as in step 4. If you have not, use the following:



```
CSQOREXX thlqual langletter
```

where langletter is a letter identifying the national language to be used:

- C** Simplified Chinese
- E** U.S. English (mixed case)
- F** French
- K** Japanese
- U** U.S. English (uppercase)

## Updating the ISPF menu

You can update the ISPF main menu to allow access to the IBM MQ operations and control panels from ISPF. The required setting for &ZSEL is:

```
CMD(%CSQOREXX thlqual langletter)
```

For information about thlqual and langletter, see Step “5” on page 936.

For more details, see the [z/OS: ISPF Dialog Developer's Guide and Reference](#).

## Updating the function keys and command settings

You can use the normal ISPF procedures for changing the function keys and command settings used by the panels. The application identifier is CSQO.

However, this is not recommended because the help information is not updated to reflect any changes that you have made.

### Related concepts

“Include the IBM MQ dump formatting member” on page 937

To be able to format IBM MQ dumps using the Interactive Problem Control System (IPCS), you must update some system libraries.

### **Include the IBM MQ dump formatting member**

To be able to format IBM MQ dumps using the Interactive Problem Control System (IPCS), you must update some system libraries.

- You need to perform this task once for each z/OS system where you want to run IBM MQ.
- You need to perform this task when migrating from a previous version.

To be able to format IBM MQ dumps using the Interactive Problem Control System (IPCS), copy the data set thlqual.SCSQPROC(CSQ7IPCS) to SYS1.PARMLIB. You should not need to edit this data set.

If you have customized the TSO procedure for IPCS, thlqual.SCSQPROC(CSQ7IPCS) can be copied into any library in the IPCSPARM definition. See [z/OS MVS IPCS User's Guide](#) for more information.

You must also include the library thlqual.SCSQPNLA in your ISPLIB concatenation.

To make the dump formatting programs available to your TSO session or IPCS job, you must also include the library thlqual.SCSQAUTH in your STEPLIB concatenation or activate it using the TSO TSOLIB command (even if it is already in the link list or LPA).

### Related concepts

[“Suppress information messages” on page 938](#)

Your IBM MQ system might produce a large number of information messages. You can prevent selected messages being sent to the console or to the hardcopy log.

### **Suppress information messages**

Your IBM MQ system might produce a large number of information messages. You can prevent selected messages being sent to the console or to the hardcopy log.

- *You need to perform this task once for each z/OS system where you want to run IBM MQ.*
- *You do not need to perform this task when migrating from a previous version.*

If your IBM MQ system is heavily used, with many channels stopping and starting, a large number of information messages are sent to the z/OS console and hardcopy log. The IBM MQ - IMS bridge and buffer manager might also produce a large number of information messages.

If required, you can suppress some of these console messages by using the z/OS message processing facility list, specified by the MPFLSTxx members of SYS1.PARMLIB. The messages you specify still appear on the hardcopy log, but not on the console.

Sample th1qua1.SCSQPROC(CSQ4MPFL) shows suggested settings for MPFLSTxx. See [MPFLSTxx \(message processing facility list\)](#) for more information.

If you want to suppress selected information messages on the hardcopy log, you can use the z/OS installation exit IEAVMXIT. You can set the following bit switches ON for the required messages:

#### **CTXTRDTM**

Delete the message.

The message is not displayed on consoles or logged in hardcopy.

#### **CTXTESJL**

Suppress from job log.

The message does not go into the JES job log.

#### **CTXTNWTP**

Do not carry out WTP processing.

The message is not sent to a TSO terminal or to the system message data set of a batch job.

#### **Note:**

1. For full details on the other parameters, see [MVS Installation Exits](#).
2. You are not recommended to suppress messages other than those in the suggested suppression list, CSQ4MPFL.

In addition you can specify the extra parameter:

#### **EXCLMSG**

Specifies a list of messages to be excluded from any log.

Messages in this list are not sent to the z/OS console and hardcopy log. See [EXCLMSG](#) in [“Utilizzo di CSQ6SYSP” on page 912](#) for further information.

### Related tasks

[“Testing a queue manager on z/OS” on page 953](#)

When you have customized or migrated your queue manager, you can test it by running the installation verification programs and some of the sample applications shipped with IBM MQ for z/OS.

## Configuring the queue sharing group

If you want to use shared queues for high availability, use these topics as a step by step guide for configuring the queue sharing group.

When you have completed the steps in this part of the process for setting up your IBM MQ for z/OS system, you should “[Tailor your system parameter module](#)” on page 910 to add queue sharing group data. You need to modify [CSQ6SYSP](#) to specify the QSGDATA parameter.

## Set up the Db2 environment

If you are using queue sharing groups you must create the required Db2 objects by customizing and running a number of sample jobs.

### Set up the Db2 environment

You must create and bind the required Db2 objects by customizing and running a number of sample jobs.

- Repeat this task for each Db2 data-sharing group.
- You need to perform the `bind` and `grant` steps when migrating from a previous version.
- Omit this task if you are not using queue sharing groups.



If you later want to use queue sharing groups, perform this task at that time.

IBM MQ provides two equivalent sets of jobs. Those with the CSQ45 prefix are for compatibility with earlier versions of IBM MQ and for use with IBM MQ version 11 and earlier. If you are setting up a new data-sharing group with Db2 V12 or later, you are encouraged to use the jobs with CSQ4X prefix, as these jobs exploit more recent Db2 capabilities for dynamic sizing and Universal Table Spaces (UTS).

The following steps must be performed for each new Db2 data-sharing group. All the sample JCL is in `thlqual.SCSQPROC`.

1. Customize and execute sample JCL CSQ4XCSG to create the storage group that is to be used for the IBM MQ database, table spaces, and tables.
2. Customize and execute sample JCL CSQ4XCDB to create the database to be used by all queue managers that are connecting to this Db2 data-sharing group.
3. Customize and execute sample JCL CSQ4XCTS to create the table spaces that contain the queue manager and channel initiator tables used for queue sharing groups.
4. Customize and execute sample JCL CSQ4XCTB to create the 15 Db2 tables and associated indexes. Do not change any of the row names or attributes.
5. Customize and execute sample JCL CSQ45BPL to bind the Db2 plans for the queue manager, utilities, and channel initiator.
6. Customize and execute sample JCL CSQ45GEX to grant execute authority to the plans for the user IDs that are used by the queue manager, utilities, and channel initiator. The user IDs for the queue manager and channel initiator are the user IDs under which their started task procedures run. The user IDs for the utilities are the user IDs under which the batch jobs can be submitted.

The names of the appropriate plans are shown in the following table.

User	Plans (  )	Plans (  )
Queue manager	CSQ5A 930, CSQ5C 930, CSQ5D 930, CSQ5K 930, CSQ5L 930, CSQ5M 930, CSQ5P 930, CSQ5R 930, CSQ5S 930, CSQ5T 930, CSQ5U 930, CSQ5W 930	CSQ5A 9X0, CSQ5C 9X0, CSQ5D 9X0, CSQ5K 9X0, CSQ5L 9X0, CSQ5M 9X0, CSQ5P 9X0, CSQ5R 9X0, CSQ5S 9X0, CSQ5T 9X0, CSQ5U 9X0, CSQ5W 9X0
SDEFS function of the CSQUTIL batch utility	CSQ52 930	CSQ52 9X0
CSQ5PQSG and CSQJUCNV batch utilities	CSQ5B 930	CSQ5B 9X0
CSQUZAP service utility	CSQ5Z 930	CSQ5Z 9X0

In the event of a failure during Db2 setup, the following jobs can be customized and executed:

- CSQ45DTB to drop the tables and indexes.
- CSQ4XDTS to drop the table spaces.
- CSQ4XDDB to drop the database.
- CSQ4XDSG to drop the storage group.

**Note:** If these jobs fail because of a Db2 locking problem it is probably due to contention for a Db2 resource, especially if the system is being heavily used. Resubmit the jobs later. It is preferable to run these jobs when the system is lightly used or quiesced.

See [Db2 Administration](#) in *Db2 for z/OS 12.0.0* for more information about setting up Db2.

See [Pianificazione su z/OS](#) for information about Db2 table sizes.

### Related concepts

“Set up the coupling facility” on page 940

If you are using queue sharing groups, define the coupling facility structures used by the queue managers in the queue sharing group (QSG) in the coupling facility Resource Management (CFRM) policy data set, using IXCMIAPU.

### **Set up the coupling facility**

If you are using queue sharing groups, define the coupling facility structures used by the queue managers in the queue sharing group (QSG) in the coupling facility Resource Management (CFRM) policy data set, using IXCMIAPU.

See [Administrative data utility](#) for more information on IXCMIAPU.

- Repeat this task for each queue sharing group.
- You might need to perform this task when migrating from a previous version.
- Omit this task if you are not using queue sharing groups.

If you later want to use queue sharing groups, perform this task at that time.

All the structures for the queue sharing group start with the name of the queue sharing group. Define the following structures:

- An administrative structure called *qsg-name* CSQ\_ADMIN. This structure is used by IBM MQ itself and does not contain any user data.
- A system application structure called *qsg-name* CSQSYSAPPL. This structure is used by IBM MQ system queues to store state information.
- One or more structures used to hold messages for shared queues. These can have any name you choose up to 16 characters long.
  - The first four characters must be the queue sharing group name. (If the queue sharing group name is less than four characters long, it must be padded to four characters with @ symbols.)
  - The fifth character must be alphabetic and subsequent characters can be alphabetic or numeric. This part of the name (without the queue sharing group name) is what you specify for the CFSTRUCT name when you define a shared queue, or a CF structure object.

You can use only alphabetic and numeric characters in the names of structures used to hold messages for shared queues, you cannot use any other characters (for example, the \_ character, which is used in the name of the administrative structure).

Sample control statements for IXCMIAPU are in data set thlqual.SCSQPROC(CSQ4CFRM). Customize these and add them to your IXCMIAPU job for the coupling facility and run it.

When you have defined your structures successfully, activate the CFRM policy that is being used. To do this, issue the following z/OS command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

For information about planning CF structures and their sizes, see [Defining coupling facility resources](#).

### Related concepts

[“Implement your ESM security controls” on page 902](#)

Implement security controls for queue managers and the channel initiator.

## **Set up the SMDS environment**

If you want to use SMDS to offload messages on shared queues, set up the SMDS offload storage environment.

- *Perform this task for each queue manager and structure in the queue sharing group that you want to configure to offload data to SMDS.*
- *If you want to configure additional structures to offload data to SMDS later, this task can be performed again at that time.*
- *Omit this task if you are not using queue sharing groups.*

*If you later want to use queue sharing groups, perform this task at that time.*

### Set up the SMDS environment

1. Estimate structure and data set space requirements. See [Shared message data set capacity considerations](#).
2. Allocate and preformat data sets. See [Creating a shared message data set](#).
3. When you define the CF structure to IBM MQ, ensure that you define the CFSTRUCT with CFLEVEL(5) and OFFLOAD(SMDS).

### Related concepts

[“Set up the coupling facility” on page 940](#)

If you are using queue sharing groups, define the coupling facility structures used by the queue managers in the queue sharing group (QSG) in the coupling facility Resource Management (CFRM) policy data set, using IXCMIAPU.

## **Add the IBM MQ entries to the Db2 tables**

If you are using queue sharing groups, run the CSQ5PQSG utility to add queue sharing group and queue manager entries to the IBM MQ tables in the Db2 data-sharing group.

- Repeat this task for each IBM MQ queue sharing group and each queue manager.
- You might need to perform this task when migrating from a previous version.
- Omit this task if you are not using queue sharing groups.

*If you later want to use queue sharing groups, perform this task at that time.*

Run CSQ5PQSG for each queue sharing group and each queue manager that is to be a member of a queue sharing group.

Perform the following actions in the specified order:

1. Add a queue sharing group entry into the IBM MQ Db2 tables using the ADD QSG function of the CSQ5PQSG program. A sample is provided in thlqual.SCSQPROC(CSQ45AQS).

Perform this function once for each queue sharing group that is defined in the Db2 data-sharing group. The queue sharing group entry must exist before adding any queue manager entries that reference the queue sharing group.

2. Add a queue manager entry into the IBM MQ Db2 tables using the ADD QMGR function of the CSQ5PQSG program. A sample is provided in thlqual.SCSQPROC(CSQ45AQM).

Perform this function for each queue manager that is to be a member of the queue sharing group.

### **Note:**

- a. A queue manager can only be a member of one queue sharing group.
- b. You must have RRS running to be able to use queue sharing groups.

### **Related concepts**

[“Tailor your system parameter module” on page 910](#)

The IBM MQ system parameter module controls the logging, archiving, tracing, and connection environments that IBM MQ uses in its operation. A default module is supplied. You should create your own system parameter module as some parameters, for example data set names, are usually site specific.

## **Implement ESM security controls for the queue sharing group**

Implement security controls for all queue managers in a queue sharing group, to access Db2 and the coupling facility list structures.

- Repeat this task for each IBM MQ queue manager in a queue sharing group.
- You might need to perform this task when migrating from a previous version.

Ensure that the user IDs associated with the queue manager, channel initiator, and the utilities have authority to establish an RRSF connection to each Db2 subsystem with which you want to establish a connection. The user IDs for the queue manager and channel initiator are the user IDs under which their started task procedures run.

The user IDs for the utilities are the user IDs under which the batch jobs can be submitted. The RACF profile to which the user ID requires READ access is Db2ssid. RRSF in the DSNR resource class

The user IDs associated with each queue manager in a queue sharing group need to be granted the appropriate level of access to the coupling facility list structures. The RACF class is FACILITY.

The following user IDs require ALTER access:

- The queue manager ID to the IXLSTR. structure-name profile
- The user ID running CSQ5PQSG

### **Related concepts**

[“Implement your ESM security controls” on page 902](#)

Implement security controls for queue managers and the channel initiator.

## **Configuring Advanced Message Security for z/OS**

Use these topics as a step by step guide for configuring Advanced Message Security (AMS).

### **Before you begin**

Before you start to configure AMS, ensure that the following queue manager configuration steps have been performed:

1. Add the CSQ0DRTM module to the LPA, as described in [“Aggiornare l'elenco di link z/OS e LPA” on page 891.](#)
2. Add an entry for CSQ0DSRV to the z/OS program properties table (PPT), as described in [“Update the z/OS program properties table” on page 895.](#)
3. Include the CSQ4INSM member in the CSQINP2 concatenation of queue manager started task procedure, as described in [“Customize the initialization input data sets” on page 903.](#)
4. Enable AMS using the AMSPROD attribute. See [product usage recording with IBM MQ for z/OS products](#) for more details.

### **What to do next**

Configure policies for queues protected by AMS. Security policies are described in [Administering Advanced Message Security security policies.](#)

There are examples of AMS configurations in [Example configurations on z/OS.](#)

## **Create procedures for Advanced Message Security**

Each IBM MQ subsystem that is to be configured to use Advanced Message Security (AMS) requires a cataloged procedure to start the AMS address space. You can create your own or use the IBM-supplied procedure library.

### **Procedure**

1. Copy the sample started task procedure *thlqual.SCSQPROC(CSQ4AMSM)* to your SYS1.PROCLIB or, if you are not using SYS1.PROCLIB, your procedure library. Name the procedure *xxxxAMSM*, where *xxxx* is the name of your IBM MQ subsystem. For example, *CSQ1AMSM* would be the AMS started task procedure for queue manager *CSQ1*.
2. Make a copy for each IBM MQ subsystem that you are going to use.
3. Tailor the procedures to your requirements using the instructions in the sample procedure *CSQ4AMSM*. You can also use symbolic parameters in the JCL to allow the procedure to be modified when it is started.
4. Review and optionally change the parameters passed to the AMS task using the Language Environment® *\_CEE\_ENVFILE* file. The sample *thlqual.SCSQPROC(CSQ40ENV)* lists the supported parameters.
5. Repeat steps 1 to 4 for each IBM MQ queue manager.

### **What to do next**

[“Set up the Advanced Message Security started task user ID” on page 943](#)

## **Set up the Advanced Message Security started task user ID**

The Advanced Message Security (AMS) task requires a user ID that allows it to be known as a z/OS UNIX System Services (z/OS UNIX) process.

## About this task

In addition, the users that the task works on behalf of must also have an appropriate definition of a UNIX UID (user ID) and GID (group ID) so these users are known as z/OS UNIX System Services users. For more information on defining z/OS UNIX System Services UIDs and GIDs, see [z/OS: Security Server RACF Security Administrator's Guide](#).

Review [z/OS UNIX System Services Planning](#) to ensure that you understand the security differences between traditional UNIX security and z/OS UNIX security. This allows you to administer the Advanced Message Security task according to your installation's security policy for deploying and running privileged z/OS UNIX System Services processes.

The primary difference between traditional UNIX security and z/OS security is that the Kernel services support two levels of appropriate privileges: UNIX level and z/OS UNIX level.

Depending on your installation's security policy, the Advanced Message Security task can either run with superuser authority (uid(0)), or with its RACF identity permitted to the RACF FACILITY class BPX.DAEMON and BPX.SERVER profiles, as this task must be able to assume the RACF identity of its users.

If the latter method is used, or you have already activated the BPX.DAEMON or BPX.SERVER profiles, the Advanced Message Security task program (`thlqual.SCSQAUTH(CSQ0DSRV)`) must be located in RACF program-controlled libraries.

**Note:** Choose the user ID for this task carefully because the Advanced Message Security recipient certificates are loaded into a key ring associated with this user ID. This consideration is discussed in [Using certificates on z/OS](#).

The steps shown here describe how to set up the Advanced Message Security started task user. The steps use RACF commands as examples. If you are using a different security manager, you should use equivalent commands.

**Note:** The examples in this section assume that you have activated generic profile command processing for the RACF STARTED, FACILITY, and SURROGAT classes and generic profile checking. For more information on how RACF handles generic profiles, see [z/OS: Security Server RACF Command Language Reference](#).

## Procedure

1. Define the Advanced Message Security started task user to RACF. The examples in this section use the user ID WMQAMSM.

```
ADDUSER WMQAMSM NAME('AMS user') OMVS (UID(0)) DFLTGRP(group)
```

Select a default 'group' as appropriate to your installation standards.

**Note:** If you do not want to grant z/OS UNIX superuser authority (UID(0)), then you must permit the Advanced Message Security user ID to the BPX.DAEMON and BPX.SERVER facility class profiles:

```
PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

and the Advanced Message Security task program (`thlqual.SCSQAUTH(CSQ0DSRV)`) must be located in a RACF program-controlled library.

To make your SCSQAUTH library program controlled, you can use the following command:

```
RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-
RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)
SETROPTS WHEN(PROGRAM) REFRESH
```

You must also enable program control for the national language library (`thlqual.SCSQANLx`) that is used by the Advanced Message Security task.

2. Determine if the RACF STARTED class is active. If it is not, activate the RACF STARTED class:



```
SETOPTS CLASSACT(STARTED)
```

3. Define a started class profile for the Advanced Message Security tasks, specifying the user ID you selected or created in step 1:

```
RDEFINE STARTED qmgrAMSM.* STDATA(USER(WMQAMSM))
```

where *qmgr* is the prefix of the started task name. For example, the started task may be named CSQ1AMSM. In this case, you would substitute *qmgr*AMSM.\* with CSQ1AMSM.\*.

The AMS started tasks must be named *qmgr*AMSM.

4. Use the **SETOPTS** RACF command to refresh the in-storage RACLISTed STARTED class profiles:

```
SETOPTS RACLIST(STARTED) REFRESH
```

5. The Advanced Message Security task temporarily assumes the identity of the host user ID of the requestor during protection processing of IBM MQ messages. Therefore, it is necessary to define profiles in the SURROGAT class for each user ID that can make requests.

If the RACF SURROGAT class is active, defining a single generic profile allows the Advanced Message Security task to assume the identity of any user. The check is ignored if the SURROGAT class is not active. The SURROGAT profiles needed are described in [z/OS UNIX System Services Planning](#).

To define profiles in the SURROGAT class:

- a) Activate the RACF SURROGAT class using the RACF SETOPTS command:

```
SETOPTS CLASSACT(SURROGAT)
```

- b) Activate generic profile processing for the RACF SURROGAT class:

```
SETOPTS GENERIC(SURROGAT)
```

- c) Activate generic profile command processing for the RACF SURROGAT class:

```
SETOPTS GENCMD(SURROGAT)
```

- d) Define a generic profile in the SURROGAT class:

```
RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
```

- e) Permit the Advanced Message Security user ID to the generic SURROGAT class profile:

```
PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQAMSM) ACCESS(READ)
```

**Note:** You can define more specific profiles if you want to restrict specific users to be processed by the Advanced Message Security task, as described in [z/OS UNIX System Services Planning](#).

For example, a profile called BPX.SRV.MQUSER1 controls whether the AMS task can assume the identity of the user ID MQUSER1.

- f) Permit the Advanced Message Security user ID to the BPX.SERVER facility (if not already done in [Creating the certificates and key rings](#)):

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

- g) Use the **SETOPTS** RACF command to refresh the in-storage RACLISTed started class profiles:

```
SETROPTS RACLIST(SURROGAT) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
```

6. The Advanced Message Security task uses the facilities provided by z/OS System SSL services to open SAF-managed key rings. The underlying System Authorization Facility (SAF) that accesses the contents of the key rings is controlled by RACF, or an equivalent security manager.

This service is the IRRSDL00 (R\_datalib) callable service. This callable service is protected with the same profiles used to protect the RACF RACDCERT commands that are defined to the RACF FACILITY class. Thus, the Advanced Message Security user ID must be permitted to the profiles using these commands:

- a) If you have not already done so, define a RACF generic profile to the RACF FACILITY class that protects the RACDCERT command and the IRRSDL00 callable service:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
```

- b) Grant authority to the started task user ID to the RACF generic profile:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(WMQASM) ACC(READ)
```

Alternatively, you can grant READ access to the data service task user's keyring in the RDATA LIB class as follows:

```
PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATA LIB) ID(WMQASM) ACC(READ)
```

7. Configure resource security:

- a) The Advanced Message Security started task user requires authority to connect to the queue manager as a batch application.

If your queue manager has connection security enabled, grant the AMS task authority to connect to the queue manager with this command:

```
PERMIT hlq.BATCH CLASS(MQCONN) ID(WMQASM) ACC(READ)
```

where *hlq* can be either the queue manager name queue sharing group name.

For further information, see [Connection security profiles for batch connections](#).

- b) The Advanced Message Security started task user requires authority to browse the SYSTEM.PROTECTION.POLICY.QUEUE.

If queue security is active on the queue manager, grant the AMS user authority to access the queue with these commands:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE) ID(WMQASM) ACCESS(READ)
```

where *hlq* can be either the queue manager name queue sharing group name.

If the queue manager is using mixed case profiles, define the profile in the MXQUEUE class instead.

To manage AMS security policies using the CSQ0UTIL utility, administrators need access to put messages to the SYSTEM.PROTECTION.POLICY.QUEUE. This is performed by granting UPDATE access to the profile protecting the queue.

For further information, see [Profiles for queue security](#).

## What to do next

[“Grant RACDCERT permissions to the security administrator for Advanced Message Security” on page 947](#)

### **Grant RACDCERT permissions to the security administrator for Advanced Message Security**

Your Advanced Message Security security administrator requires authority to use the RACDCERT command to create and manage digital certificates.

## Procedure

- Identify the appropriate user ID for this role and grant permission to use the RACDCERT command. For example:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

where admin is the user ID of your Advanced Message Security security administrator.

## What to do next

[“Grant users resource permissions for Advanced Message Security” on page 947](#)

### **Grant users resource permissions for Advanced Message Security**

Advanced Message Security users require relevant resource permissions.

## About this task

Advanced Message Security users, that is users that are putting or getting Advanced Message Security protected messages, require:

- An OMVS segment associated with their user id
- Permissions for IRR.DIGTCERT.LISTRING or RDATA LIB
- Permissions for ICSF class CSFSERV and CSFKEYS profiles
- Permission to put to the SYSTEM.PROTECTION.ERROR.QUEUE

The Advanced Message Security task temporarily assumes the identity of its clients; that is, the task acts as a surrogate of the z/OS user ID of users of Advanced Message Security during the processing of IBM MQ messages to queues that are protected by Advanced Message Security.

In order for the task to assume the z/OS identity of a user, the client z/OS user ID must have a defined OMVS segment associated with its user profile.

As an administration aid, RACF provides the ability to define a default OMVS segment that may be associated with RACF user and group profiles. This default is used if the z/OS user ID or group profile does not have an OMVS segment explicitly defined. If you plan to have a large number of users using Advanced Message Security, you might choose to use this default rather than explicitly defining the OMVS segment for each user.

The *z/OS: Security Server RACF Security Administrator's Guide* contains the detailed procedure for defining default OMVS segments. Review the procedure as outlined in this publication to determine if the definition of default OMVS segments in RACF User and Group profiles is appropriate to your installation.

## Procedure

1. Grant READ permission to the IRR.DIGTCERT.LISTRING profile in the FACILITY class:
  - To grant READ permission to the IRR.DIGTCERT.LISTRING profile in the FACILITY class to all users, issue this command:

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
```

- To grant READ permission to the IRR.DIGTCERT.LISTRING profile in the FACILITY class on a per user basis, issue this command:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

where *userid* is the name of the Advanced Message Security user.

- Alternatively, use the RDATALIB class to grant access to specific key rings. The RDATALIB permissions take precedence over IRR.DIGTCERT.LISTRING permissions. For example:

```
PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(user) ACC(READ)
```

2. If you are using ICSF-managed certificates and private keys, Advanced Message Security users require access to certain class CSFSERV and CSFKEYS profiles. This access is detailed in the following table:

Class	Profile	Permission
CSFSERV	CSFDSG	READ
CSFSERV	CSFPKE	READ
CSFSERV	CSFPKD	READ
CSFSERV	CSFDSV	READ
CSFKEYS	ICSF PKDS Label	READ

3. Applications that perform operations on queues with AMS policies defined need access to put messages to SYSTEM.PROTECTION.ERROR.QUEUE. Grant put access to the queue with these commands:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
PERMIT hlq.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE) ID(userID) ACCESS(UPDATE)
```

where *hlq* can be either the queue manager name queue sharing group name, and *userID* is the application user ID.

## What to do next

[“Create key rings for Advanced Message Security” on page 948](#)

### Create key rings for Advanced Message Security

Certificates used by Advanced Message Security (AMS) for signing and encryption are stored in z/OS SAF key rings. You need to create these key rings and certificates before you can use AMS.

## About this task

Advanced Message Security accesses certificates in the following key rings:

- A single key ring owned by the AMS address space user.
- Key rings owned by the individual users that send or receive messages on queues with AMS policies defined.

These key rings must all be named `drq.ams.keyring`.

There is more information on key rings and certificates used by AMS, and an example scenario, in [Using certificates on z/OS](#).

Follow these steps to create the key rings required by AMS, and connect certificates to the key rings. You must create the key ring owned by the AMS address space user before starting AMS. You can create the keys rings owned by the users that send or receive messages at any time.

## Procedure

1. Issue the following command to create a key ring owned by the AMS address space user:

```
RACDCERT ID(amsUser) ADDRING(dmq.ams.keyring)
```

where *amsUser* is the user ID of the AMS address space.

2. Create a key ring for each user that sends or receives messages protected by AMS by issuing the command in step 1 for each user ID.
3. Connect the certificate authority (CA) certificate for the issuer of the user certificates to the key ring owned by the AMS address space user ID. Issue the following command:

```
RACDCERT ID(amsUser) CONNECT(CERTAUTH LABEL('caLabel') RING(dmq.ams.keyring))
```

where *amsUser* is the user ID of the AMS address space, and *caLabel* is the label of the CA certificate.

If you are using RACF as your CA, and need to create a certificate authority certificate, follow the example in [Defining a local Certificate Authority certificate](#).

4. If you are using privacy or confidentiality security policies to encrypt messages on queues protected by AMS, connect the certificates of message recipients to the key ring owned by the AMS address space user ID. Issue the following command:

```
RACDCERT ID(amsUser) CONNECT(ID(userId) LABEL('certLabel')
RING(dmq.ams.keyring) USAGE(SITE))
```

where *amsUser* is the user ID of the AMS address space, *userId* is the message recipient, and *certLabel* is the label of the user's certificate.

The USAGE(SITE) attribute prevents the private key from being accessible in the key ring.

If you are creating your own certificates with RACF, follow the example in [Creating a digital certificate with a private key](#) to create the certificate.

5. Connect the certificates of each user that sends or receives messages protected by AMS to a key ring owned by the user. The certificate must be connected as the default certificate in the key ring. Issue the following command:

```
RACDCERT ID(userId) CONNECT(ID(userId) LABEL('certLabel')
RING(dmq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

where *userId* is the user that is sending or receiving messages, and *certLabel* is the label of the user's certificate.

### Notes:

- a. Steps “2” on page 949 and “5” on page 949 are not required if the application opens a queue only for output, and sends messages to queues protected by an AMS confidentiality policy.
- b. Steps “2” on page 949 and “5” on page 949 are not required if the application opens a queue only for input/browse, and receives messages from queues protected by an AMS integrity policy.

## What to do next

[“Enable Advanced Message Security” on page 949](#)

### **Enable Advanced Message Security**

Security policy capability for a queue manager is controlled by the SPLCAP parameter in the system parameter module.

## About this task

Follow these steps to enable Advanced Message Security (AMS) for a single queue manager.

This task requires you to make a change to the system parameter module. See [“Tailor your system parameter module”](#) on page 910 for more information on creating and customizing the system parameter module.

## Procedure

1. Set **SPLCAP** to YES in CSQ6SYSP. See [“Utilizzo di CSQ6SYSP”](#) on page 912 for more information on the CSQ6SYSP macro.
2. Set the **AMSPROD** to either AMS, ADVANCED, or ADVANCEDVUE depending on your licence entitlement. See [using CSQ6USGP](#) for more information on the CSQ6USGP macro.
3. Recompile the system parameter module.
4. Restart the queue manager with the updated system parameter module. The AMS address space is started automatically when the queue manager starts.

## **Configuring the mqweb server**

Use these topics as a step by step guide for configuring the mqweb server.

### Related tasks

[“Configurazione di IBM MQ Console e REST API”](#) on page 837

Il server mqweb che ospita IBM MQ Console e REST API viene fornito con una configurazione predefinita. Per poter utilizzare uno di questi componenti è necessario completare una serie di attività di configurazione, come la configurazione della sicurezza per consentire agli utenti di accedere. Questo argomento descrive tutte le opzioni di configurazione disponibili.

## **Creazione del server mqweb**

Se è stato installato IBM MQ for z/OS UNIX System Services Web Componentse si desidera utilizzare IBM MQ Consoleo REST API, è necessario creare e personalizzare il server mqweb.

## Prima di iniziare

Prima di eseguire lo script **crtmqweb** per creare il server mqweb, impostare la variabile di ambiente JAVA\_HOME in modo che faccia riferimento a una versione a 64 bit di Java sul sistema.

IBM MQ Console e administrative REST API richiedono SYSTEM.REST.REPLY.QUEUE da creare. Creare questa coda utilizzando l'esempio **CSQ4INSG** in [“Customize the initialization input data sets”](#) a pagina 903.



**Attenzione:** Quando si avvia il server mqweb, se viene visualizzato il messaggio di errore CWWKG0014E, come visualizzato nel seguente output:

```
Launching mqweb (MQM MVS/ESA V9 R2.0/wlp...) (en_US)
 YAUDIT " CWWKE0001I: The server mqweb has been
launched.
 YWARNING " CWWKF0009W: The server has not been configured to install any
features.
 YAUDIT " CWWKF0011I: The mqweb server is ready to run a smarter planet.
The mqweb server started in 6.348 seconds.
 YERROR " CWWKG0014E: The configuration parser detected an XML syntax
error while parsing the root of the configuration and the referenced configuration
documents.
 Error: An invalid XML character (Unicode: 0x4c) was found
in the prolog of the document.
 File: file:<your filepath>/servers/mqweb/server.xml Line:
1 Column: 1
```

è necessario controllare l'impostazione z/OS di AUTOCVT (convertire automaticamente i file da una serie di codici a un altro) e regolare il valore come richiesto effettuando una delle seguenti operazioni.

**In un terminale USS:**

Immettere il comando: `echo $_BPXK_AUTOCVT` per visualizzare il valore di questa variabile di ambiente. Se la variabile di ambiente non è definita, non viene visualizzato alcun valore.

Per impostare la variabile di ambiente, vedi [\\_BPXK environment variables](#).

**A livello di sistema:**

L'esempio 6 in [Visualizzazione dello stato di z/OS UNIX System Services \(OMVS\)](#) mostra come visualizzare il valore dell'istruzione AUTOCVT del sistema in BPXPRMxx.

Per impostare la variabile di ambiente a livello di sistema, utilizzare l'istruzione [AUTOCVT](#) in BPXPRMxx.

Se la variabile di ambiente `_BPccedenze AUTOCVT` è impostata in un terminale USS, sovrascrive l'impostazione a livello di sistema dell'istruzione AUTOCVT in BPXPRMxx.

## Informazioni su questa attività

- Completare questa attività una volta per ogni sistema z/OS in cui si desidera eseguire IBM MQ Console o REST API.
- Per utilizzare amministrative REST API, è necessario un server mqweb per ciascuna versione di IBM MQ in esecuzione. Ad esempio, se si stanno eseguendo IBM MQ 9.4.0, 9.3.5 e 9.3.0, sono necessari tre diversi server mqweb.
- Potrebbe essere necessario aggiornare o modificare la configurazione del server quando si esegue la migrazione da una versione precedente.

IBM MQ Console e REST API richiedono la creazione di un singolo server WebSphere Liberty , denominato mqweb.

I file di log e di configurazione del server sono tutti memorizzati nella directory utente Liberty .

Il server mqweb deve essere configurato con un ID prodotto (PID) con cui viene eseguito. Il PID viene impostato quando viene creato il server mqweb. Utilizzare lo stesso PID utilizzato per eseguire i gestori code locali a cui si connette il server mqweb.

**Nota:** Se i gestori code locali vengono eseguiti in più PID differenti, scegliere uno di essi in cui eseguire il server mqweb.

Per ulteriori informazioni sui PID e su come vengono utilizzati in z/OS, vedi [Registrazione dell'utilizzo del prodotto con prodotti IBM MQ for z/OS](#).

È possibile modificare il PID con cui viene eseguito il server mqweb, dopo che è stato creato, utilizzando il comando `setmqweb pid` .

Completare la seguente procedura per creare il server mqweb:

## Procedura

1. Decidere con quale PID viene eseguito il server mqweb.

2. Scegliere un percorso adatto per la directory utente Liberty .

L'ID utente con cui viene eseguito il server mqweb, richiede l'accesso in lettura e scrittura a questa directory utente e al relativo contenuto. Poiché questa directory utente contiene file di log, in aggiunta alla configurazione del server, creare questa directory in un file system separato.

**Nota:** C'è una quantità significativa di I/O del disco all'avvio del server mqweb. Per ridurre il tempo impiegato per avviare il server mqweb, assicurarsi che sia il file system di IBM MQ installazione z/OS UNIX che il file system della directory utente Liberty siano sysplex-aware o montati localmente sul sistema su cui è in esecuzione il server mqweb.

3. In z/OS UNIX System Services, modificare la propria directory di lavoro corrente in `PathPrefix/web/bin` immettendo il seguente comando:

```
cd PathPrefix/web/bin
```

dove *PathPrefix* è il percorso di installazione IBM MQ for z/OS UNIX System Services Components .

4. Creare la directory utente Liberty che contiene la definizione del server mqweb modello, eseguendo lo script **crtmqweb** .

Il formato del comando **crtmqweb** è:

```
crtmqweb user_directory -p pid_value
```

dove:

#### **directory\_utente**

È la directory utente Liberty decisa nel passo “2” a pagina 951. Questo parametro è facoltativo.

Se questo parametro non viene specificato, viene utilizzata una directory utente Liberty predefinita di `/var/mqm/web/installation1` .

#### **valore\_pid**

Indica il PID con cui viene eseguito il server mqweb. Questo PID è quello scelto nel passaggio “1” a pagina 951. *pid\_value* è uno dei seguenti valori:

##### **MQ**

Il server mqweb viene eseguito in PID IBM MQ for z/OS (5655-MQ9).

##### **VUE**

Il server mqweb viene eseguito con PID IBM MQ for z/OS Value Unit Edition (VUE) (5655-VU9).

##### **AVANZATOVUE**

Il server mqWeb viene eseguito con PID IBM MQ Advanced for z/OS VUE (5655-AV1),

Ad esempio, per creare il server mqweb con una directory utente Liberty di `/usr/mqweb` e un PID di IBM MQ Advanced for z/OS VUE (5655-AV1), eseguire il seguente comando:

```
./crtmqweb /usr/mqweb -p ADVANCEDVUE
```

5. Modificare la proprietà delle directory e dei file nella directory utente Liberty , in modo che appartengano all'ID utente e al gruppo con cui viene eseguito il server mqweb, utilizzando il seguente comando:

```
chown -R userid:group path
```

Per fornire al gruppo l'accesso in scrittura al percorso, immettere il comando:

```
chmod -R 770 path
```

## **Operazioni successive**

[“Creating a procedure for the mqweb server” a pagina 952](#)

### **Attività correlate**

[“Configurazione di IBM MQ Console e REST API” a pagina 837](#)

Il server mqweb che ospita IBM MQ Console e REST API viene fornito con una configurazione predefinita. Per poter utilizzare uno di questi componenti è necessario completare una serie di attività di configurazione, come la configurazione della sicurezza per consentire agli utenti di accedere. Questo argomento descrive tutte le opzioni di configurazione disponibili.

### **z/OS Creating a procedure for the mqweb server**

If you installed the IBM MQ for z/OS UNIX System Services Web Components, and want to use the IBM MQ Console, or the REST API, you need to create a cataloged procedure to start the mqweb server. The mqweb server is a Liberty server that hosts the IBM MQ Console and the REST API.

- You need to perform this task once for each z/OS system where you want to run the IBM MQ Console or REST API.
- You need a mqweb server for each version of IBM MQ that is running. For example, a started task called MQWB0910 for queue managers at IBM MQ for z/OS 9.1.0 and a started task called MQWB0905 for queue managers at IBM MQ for z/OS 9.0.5.



If you have only one queue manager on the z/OS system, you can run a single Liberty server started task, and change the libraries it uses when you migrate your queue manager.

- You might need to modify the cataloged procedure when migrating from a previous version.

Carry out the following procedure to create a cataloged procedure:

1. Copy the sample started task procedure `th1qua1.SCSQPROC (CSQ4WEBS)` to your procedure library.

Name the procedure according to the standards of your enterprise.

For example `MQWB0910`, indicating that this is the cataloged procedure for the IBM MQ for z/OS 9.1.0 mqweb server.

2. Tailor the procedure to your requirements using the instructions in the sample procedure `CSQ4WEBS`.

Note that the Liberty user directory is the directory specified when the `crtmqweb` script was run to create the mqweb server definition.

See [“Creazione del server mqweb” on page 950](#) for details.

**Note:** Ensure that you specify **Caps off** when you edit the member, as the file has lowercase data.

3. Authorize the procedure to run under your external security manager.

4. Use IBM Workload Manager (WLM) to classify this address space.

The mqweb server is an IBM MQ application, and users interact with this application. The application does not need to be high importance in WLM, and a service class of **STCUSER** might be suitable.

## What to do next

Follow the steps in [“Configurazione di base per il server mqweb” on page 838](#) to finish configuring the mqweb server.

### Related tasks

[“Configurazione di IBM MQ Console e REST API” on page 837](#)

Il server mqweb che ospita IBM MQ Console e REST API viene fornito con una configurazione predefinita. Per poter utilizzare uno di questi componenti è necessario completare una serie di attività di configurazione, come la configurazione della sicurezza per consentire agli utenti di accedere. Questo argomento descrive tutte le opzioni di configurazione disponibili.

▶ z/OS

## Testing a queue manager on z/OS

When you have customized or migrated your queue manager, you can test it by running the installation verification programs and some of the sample applications shipped with IBM MQ for z/OS.

### About this task

After you have installed and customized IBM MQ for z/OS, you can use the supplied installation verification program, `CSQ4IVP1`, to confirm that IBM MQ for z/OS is operational.

The basic installation verification program `CSQ4IVP1` tests non-shared queues and verifies the base IBM MQ without using the C, COBOL, or CICS samples.

After running the basic installation verification, you can test for shared queues by using `CSQ4IVP1` with different queues, and also test that Db2 and the coupling facility are set up correctly. To confirm that distributed queuing is operational, you can use the supplied installation verification program, `CSQ4IVPX`,

`CSQ4IVP1` is supplied as a load module, and provides a set of procedural sample applications as source modules that demonstrate typical uses of the Message Queue Interface (MQI). You can use these source modules to test different programming language environments. You can compile and link-edit whichever of the other samples are appropriate to your installation by using the supplied sample JCL supplied.

## Procedure

- For information on how to test your queue manager on z/OS, see the following subtopics:
  - [“Running the basic installation verification program” on page 954](#)
  - [“Testing for queue sharing groups” on page 957](#)
  - [“Testing for distributed queuing” on page 958](#)
  - [“Testing for C, C++, COBOL, PL/I, and CICS programs with IBM MQ for z/OS” on page 961](#)

## Related concepts

[IBM MQ for z/OS concepts](#)

## Related tasks

[Planning your IBM MQ environment on z/OS](#)

[“Configuring queue managers on z/OS” on page 883](#)

Use these instructions to configure queue managers on IBM MQ for z/OS.

[Administering IBM MQ for z/OS](#)

## **Running the basic installation verification program**

After you have installed and customized IBM MQ, you can use the supplied installation verification program, CSQ4IVP1, to confirm that IBM MQ is operational.

The basic installation verification program is a batch assembler IVP that verifies the base IBM MQ without using the C, COBOL, or CICS samples.

The Batch Assembler IVP is link-edited by SMP/E and the load modules are shipped in library thlqual.SCSQLOAD.

After you have completed both the SMP/E APPLY step and the customization steps, run the Batch Assembler IVP.

See these sections for further details:

- [Overview of the CSQ4IVP1 application](#)
- [Preparing to run CSQ4IVP1](#)
- [Running CSQ4IVP1](#)
- [Checking the results of CSQ4IVP1](#)

## **Overview of the CSQ4IVP1 application**

CSQ4IVP1 is a batch application that connects to your IBM MQ subsystem and performs these basic functions:

- Issues IBM MQ calls
- Communicates with the command server
- Verifies that triggering is active
- Generates and deletes a dynamic queue
- Verifies message expiry processing
- Verifies message commit processing

## **Preparing to run CSQ4IVP1**

Before you run CSQ4IVP1:

1. Check that the IVP entries are in the CSQINP2 data set concatenation in the queue manager startup program. The IVP entries are supplied in member thlqual.SCSQPROC(CSQ4IVPQ). If not, add the

definitions supplied in thlqual.SCSQPROC(CSQ4IVPQ) to your CSQINP2 concatenation. If the queue manager is currently running, you need to restart it so that these definitions can take effect.

2. The sample JCL, CSQ4IVPR, required to run the installation verification program is in library thlqual.SCSQPROC.

Customize the CSQ4IVPR JCL with the high-level qualifier for the IBM MQ libraries, the national language you want to use, the four-character IBM MQ queue manager name, and the destination for the job output.

3. Update RACF to allow CSQ4IVP1 to access its resources if IBM MQ security is active.

To run CSQ4IVP1 when IBM MQ security is enabled, you need a RACF user ID with authority to access the objects. For details of defining resources to RACF, see [Setting up security on z/OS](#). The user ID that runs the IVP must have the following access authority:

Authority	Profile	Class
READ	ssid.DISPLAY.PROCESS	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
UPDATE	ssid.CSQ4IVP1.**	MQQUEUE
READ	ssid.BATCH	MQCONN

These requirements assume that all IBM MQ security is active. The RACF commands to activate IBM MQ security are shown in [Figure 98 on page 955](#). This example assumes that the queue manager name is CSQ1 and that the user ID of the person running sample CSQ4IVP1 is TS101.

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

*Figure 98. RACF commands for CSQ4IVP1*

## Running CSQ4IVP1

When you have completed these steps, start your queue manager. If the queue manager is already running and you have changed CSQINP2, you must stop the queue manager and restart it.

The IVP runs as a batch job. Customize the job card to meet the submission requirements of your installation.


## Checking the results of CSQ4IVP1

The IVP is split into 10 stages; each stage must complete with a zero completion code before the next stage is run. The IVP generates a report, listing:

- The name of queue manager that is being connected to.

- A one-line message showing the completion code and the reason code returned from each stage.
- A one-line informational message where appropriate.

A sample report is provided in [Figure 99 on page 957](#)

 For an explanation of the completion and reason codes, see the [IBM MQ for z/OS messages, completion, and reason codes](#).

Some stages have more than one IBM MQ call and, in the event of failure, a message is issued indicating the specific IBM MQ call that returned the failure. Also, for some stages the IVP puts explanatory and diagnostic information into a comment field.

The IVP job requests exclusive control of certain queue manager objects and therefore should be single threaded through the system. However, there is no limit to the number of times the IVP can be run against your queue manager.

The functions performed by each stage are:

### Stage 1

Connect to the queue manager by issuing the MQCONN API call.

### Stage 2

Determine the name of the system-command input queue used by the command server to retrieve request messages. This queue receives display requests from Stage 5.

To do this, the sequence of calls is:

1. Issue an MQOPEN call, specifying the queue manager name, to open the queue manager object.
2. Issue an MQINQ call to find out the name of the system-command input queue.
3. Issue an MQINQ call to find out about various queue manager event switches.
4. Issue an MQCLOSE call to close the queue manager object.

On successful completion of this stage, the name of the system-command input queue is displayed in the comment field.

### Stage 3

Open an initiation queue using an MQOPEN call.

This queue is opened at this stage in anticipation of a trigger message, which arrives as a result of the command server replying to the request from Stage 5. The queue must be opened for input to meet the triggering criteria.

### Stage 4

Create a permanent dynamic queue using the CSQ4IVP1.MODEL queue as a model. The dynamic queue has the same attributes as the model from which it was created. This means that when the replies from the command server request in Stage 5 are written to this queue, a trigger message is written to the initiation queue opened in Stage 3.

Upon successful completion of this stage, the name of the permanent dynamic queue is indicated in the comment field.

### Stage 5

Issue an MQPUT1 request to the command server command queue.

A message of type MQMT\_REQUEST is written to the system-command input queue requesting a display of process CSQ4IVP1. The message descriptor for the message specifies the permanent dynamic queue created in Stage 4 as the reply-to queue for the command server's response.

### Stage 6

Issue an MQGET request from the initiation queue. At this stage, a GET WAIT with an interval of 1 minute is issued against the initiation queue opened in Stage 3. The message returned is expected to be the trigger message generated by the command server's response messages being written to the reply-to queue.

## Stage 7

Delete the permanent dynamic queue created in Stage 4. As the queue still has messages on it, the MQCO\_PURGE\_DELETE option is used.

## Stage 8

1. Open a dynamic queue.
2. MQPUT a message with an expiry interval set.
3. Wait for the message to expire.
4. Attempt to MQGET the expired message.
5. MQCLOSE the queue.

## Stage 9

1. Open a dynamic queue.
2. MQPUT a message.
3. Issue MQCMIT to commit the current unit of work.
4. MQGET the message.
5. Issue MQBACK to backout the message.
6. MQGET the same message and ensure that the backout count is set to 1.
7. Issue MQCLOSE to close the queue.

## Stage 10

Disconnect from the queue manager using **MQDISC**.

After running the IVP, you can delete any objects that you no longer require.

If the IVP does not run successfully, try each step manually to find out which function is failing.

```
DATE : 2005.035 IBM MQ for z/OS - V6 PAGE : 0001
INSTALLATION VERIFICATION PROGRAM
PARAMETERS ACCEPTED. PROGRAM WILL CONNECT TO : CSQ1
,OBJECT QUALIFER : CSQ4IVP1
INSTALLATION VERIFICATION BEGINS :
STAGE 01 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR BRIDGE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS EXCP FOR CHANNEL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR SSL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR INHIBITED EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR LOCAL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR PERFORMANCE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR REMOTE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR START/STOP EVENTS
STAGE 02 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 SYSTEM.COMMAND.INPUT
STAGE 03 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 04 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810EFEAC8980
STAGE 05 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 06 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 07 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 08 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810F0070E645
STAGE 09 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9812BA8706803
STAGE 10 COMPLETE. COMPCODE : 0000 REASON CODE : 0000>>>>>>>>>> END OF REPORT <<<<<<<<<<<<
```

Figure 99. Sample report from CSQ4IVP1

## Testing for queue sharing groups

The basic installation verification program CSQ4IVP1 tests non-shared queues.

CSQ4IVP1 can be used whether the queue manager is a member of a queue sharing group or not. After running the basic IVP, you can test for shared queues by using the CSQ4IVP1 installation verification program with different queues. Also this tests that Db2 and the coupling facility are set up correctly.

## Preparing to run CSQ4IVP1 for a queue sharing group

Before you run CSQ4IVP1:

1. Add the coupling facility structure that the IVP uses to your CFRM policy data set, as described in “[Set up the coupling facility](#)” on [page 940](#). The supplied samples use a structure called APPLICATION1, but you can change this if you want.
2. Check that the IVP entries are in the CSQINP2 data set concatenation in the queue manager startup program. The IVP entries are supplied in member thlqual.SCSQPROC(CSQ4IVPG). If they are not, add the definitions supplied in thlqual.SCSQPROC(CSQ4IVPG) to your CSQINP2 concatenation. If the queue manager is currently running, you need to restart it so that these definitions can take effect.
3. Change the name of the coupling facility structure used in thlqual.SCSQPROC(CSQ4IVPG) if necessary.
4. The sample JCL, CSQ4IVPS, required to run the installation verification program for a queue sharing group is in library thlqual.SCSQPROC.

Customize the CSQ4IVPS JCL with the high-level qualifier for the IBM MQ libraries, the national language you want to use, the four-character IBM MQ queue manager name, and the destination for the job output.

5. Update RACF to allow CSQ4IVP1 to access its resources if IBM MQ security is active.

To run CSQ4IVP1 when IBM MQ security is enabled, you need a RACF user ID with authority to access the objects. For details of defining resources to RACF, see [Setting up security on z/OS](#). The user ID that runs the IVP must have the following access authority in addition to that required to run the basic IVP:

Authority	Profile	Class
UPDATE	ssid.CSQ4IVPG.**	MQQUEUE

These requirements assume that all IBM MQ security is active. The RACF commands to activate IBM MQ security are shown in [Figure 100 on page 958](#). This example assumes that the queue manager name is CSQ1 and that the user ID of the person running sample CSQ4IVP1 is TS101.

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

*Figure 100. RACF commands for CSQ4IVP1 for a queue sharing group*

## Running CSQ4IVP1 for a queue sharing group

When you have completed these steps, start your queue manager. If the queue manager is already running and you have changed CSQINP2, you must stop the queue manager and restart it.

The IVP runs as a batch job. Customize the job card to meet the submission requirements of your installation.

## Checking the results of CSQ4IVP1 for a queue sharing group

The IVP for queue sharing groups works in the same way as the basic IVP, except that the queues that are created are called CSQIVPG. xx. Follow the instructions given in “[Checking the results of CSQ4IVP1](#)” on [page 955](#) to check the results of the IVP for queue sharing groups.

## Testing for distributed queuing

You can use the supplied installation verification program, CSQ4IVPX, to confirm that distributed queuing is operational.

## Overview of CSQ4IVPX job

CSQ4IVPX is a batch job that starts the channel initiator and issues the IBM MQ DISPLAY CHINIT command. This verifies that all major aspects of distributed queuing are operational, while avoiding the need to set up channel and network definitions.

## Preparing to run CSQ4IVPX

Before you run CSQ4IVPX:

1. The sample JCL, CSQ4IVPX, required to run the installation verification program is in library thlqual.SCSQPROC.

Customize the CSQ4IVPX JCL with the high-level qualifier for the IBM MQ libraries, the national language you want to use, the four-character queue manager name, and the destination for the job output.

2. Update RACF to allow CSQ4IVPX to access its resources if IBM MQ security is active. To run CSQ4IVPX when IBM MQ security is enabled, you need a RACF user ID with authority to access the objects. For details of defining resources to RACF, see [Setting up security on z/OS](#). The user ID that runs the IVP must have the following access authority:

Authority	Profile	Class
CONTROL	ssid.START.CHINIT and ssid.STOP.CHINIT	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CSQUTIL.*	MQQUEUE
READ	ssid.BATCH	MQCONN
READ	ssid.DISPLAY.CHINIT	MQCMDS

These requirements assume that the connection security profile ssid.CHIN has been defined (as shown in [Connection security profiles for the channel initiator](#)), and that all IBM MQ security is active. The RACF commands to do this are shown in [Figure 101 on page 960](#). This example assumes that:

- The queue manager name is CSQ1
- The user ID of the person running sample CSQ4IVPX is TS101
- The channel initiator address space is running under the user ID CSQ1MSTR

3. Update RACF to allow the channel initiator address space the following access authority:

Authority	Profile	Class
READ	ssid.CHIN	MQCONN
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
UPDATE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

The RACF commands to do this are also shown in [Figure 101 on page 960](#).

```

RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)

```

*Figure 101. RACF commands for CSQ4IVPX*

## Running CSQ4IVPX

When you have completed these steps, start your queue manager.

The IVP runs as a batch job. Customize the job card to meet the submission requirements of your installation.

## Checking the results of CSQ4IVPX

CSQ4IVPX runs the CSQUTIL IBM MQ utility to issue three MQSC commands. The SYSPRINT output data set should look like [Figure 102 on page 961](#), although details might differ depending on your queue manager attributes.

- You should see the commands **(1)** each followed by several messages.
- The last message from each command should be "CSQ9022I ... NORMAL COMPLETION" **(2)**.
- The job as a whole should complete with return code zero **(3)**.



```

CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRQDM Channel initiator active
CSQX002I +CSQ1 CSQXRQDM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRQDM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRQDM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRQDM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRQDM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRQDM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRQDM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRQDM TCP/IP system name is TCP/IP
CSQX848I +CSQ1 CSQXRQDM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRQDM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRQDM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRQDM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)

```

Figure 102. Example output from CSQ4IVPX

## **z/OS** Testing for C, C++, COBOL, PL/I, and CICS programs with IBM MQ for z/OS

You can test for C, C++, COBOL, PL/I, or CICS, using the sample applications supplied with IBM MQ.

The IVP (CSQ4IVP1) is supplied as a load module, and provides the samples as source modules. You can use these source modules to test different programming language environments.

For more information about sample applications, see [Sample programs for IBM MQ for z/OS](#).

## **z/OS** Setting up communications with other queue managers on z/OS

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

## About this task

To define your distributed-queuing requirements, you need to define the following items:

- The channel initiator procedures and data sets
- The channel definitions
- The queues and other objects
- Access security

If you are using queue sharing groups, see [Distributed queuing and queue sharing groups](#).

For additional points to consider when you are preparing to set up distributed queuing with IBM MQ for z/OS, see [“Considerations for using distributed queuing on z/OS” on page 962](#).

## Procedure

To enable distributed queuing, complete the following steps:

- Customize the distributed queuing facility and define the IBM MQ objects required as described in [Defining system objects](#) and [“Preparing to customize queue managers on z/OS” on page 884](#).
- Define access security as described in [Security considerations for the channel initiator on z/OS](#).
- Set up your communications as described in [“Setting up communication for z/OS” on page 981](#).

## Related concepts

[“Setting up IBM MQ for z/OS” on page 888](#)

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

## Related tasks

[“Configurazione dell'accodamento distribuito” on page 205](#)

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , include la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

## Considerations for using distributed queuing on z/OS

Points to consider when you are preparing to use distributed queuing on z/OS.

If you are using queue sharing groups, see [Distributed queuing and queue sharing groups](#).

## Operator messages

Because the channel initiator uses a number of asynchronously operating dispatchers, operator messages might occur on the log out of chronological sequence.

## Channel operation commands

Channel operation commands generally involve two stages. When the command syntax has been checked and the existence of the channel verified, a request is sent to the channel initiator. Message [CSQM134I](#) or [CSQM137I](#) is sent to the command issuer to indicate the completion of the first stage. When the channel initiator has processed the command, further messages indicating its success or otherwise are sent to the command issuer along with message [CSQ9022I](#) or [CSQ9023E](#). Any error messages generated could also be sent to the z/OS console.

All cluster commands except **DISPLAY CLUSQMgr**, however, work asynchronously. Commands that change object attributes update the object and send a request to the channel initiator. Commands for working with clusters are checked for syntax and a request is sent to the channel initiator. In both cases, message [CSQM130I](#) is sent to the command issuer indicating that a request has been sent. This message is followed by message [CSQ9022I](#) to indicate that the command has completed successfully, in that a request has been sent. It does not indicate that the cluster request has completed successfully. The requests sent to the channel initiator are processed asynchronously, along with cluster requests received

from other members of the cluster. In some cases, these requests must be sent to the whole cluster to determine if they are successful or not. Any errors are reported to the z/OS on the system where the channel initiator is running. They are not sent to the command issuer.

## Undelivered-message queue

A Dead Letter handler is provided with IBM MQ for z/OS. For more information, see [The dead-letter queue handler utility \(CSQUDLQH\)](#).

## Queues in use

MCAs for receiver channels can keep the destination queues open even when messages are not being transmitted. This behavior results in the queues appearing to be 'in use'.

## Security changes

If you change security access for a user ID, the change might not take effect immediately. For more information, see [Security considerations for the channel initiator on z/OS](#) , [Profiles for queue security](#), and [“Implement your ESM security controls”](#) on page 902.

## Communications stopped - TCP

If TCP is stopped for some reason and then restarted, the IBM MQ for z/OS TCP listener waiting on a TCP port is stopped.

Automatic channel-reconnect allows the channel initiator to detect that TCP/IP is unavailable and to automatically restart the TCP/IP listener when TCP/IP returns. This automatic restart alleviates the need for operations staff to notice the problem with TCP/IP and manually restart the listener. While the listener is out of action, the channel initiator can also be used to try the listener again at the interval specified by LSTRTMR. These attempts can continue until TCP/IP returns and the listener successfully restarts automatically. For more information about LSTRTMR, see [ALTER QMGR](#) and [Distributed queuing messages \(CSQX...\)](#).

## Communications stopped - LU6.2

If APPC is stopped, the listener is also stopped. Again, in this case, the listener automatically tries again at the LSTRTMR interval so that, if APPC restarts, the listener can restart too.

If the Db2 fails, shared channels that are already running continue to run, but any new channel start requests fail. When the Db2 is restored new requests are able to complete.

## z/OS Automatic Restart Management (ARM)

Automatic restart management (ARM) is a z/OS recovery function that can improve the availability of specific batch jobs or started tasks (for example, subsystems). It can therefore result in a faster resumption of productive work.

To use ARM, you must set up your queue managers and channel initiators in a particular way to make them restart automatically. For more information, see [Using the z/OS Automatic Restart Manager \(ARM\)](#).

### Related concepts

[“Setting up IBM MQ for z/OS”](#) on page 888

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

### Related tasks

[“Configurazione dell'accodamento distribuito”](#) on page 205

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

## Defining IBM MQ objects on z/OS

On z/OS, use one of the IBM MQ command input methods to define IBM MQ objects.

For more information about defining objects, see [“Monitoring and controlling channels on z/OS”](#) on page 965.

### Transmission queues and triggering channels

Define the following:

- A local queue with the usage of XMITQ for each sending message channel.
- Remote queue definitions.

A remote queue object has three distinct uses, depending upon the way the name and content are specified:


- Remote queue definition
- Queue manager alias definition
- Reply-to queue alias definition

These three ways are shown in [Three ways of using the remote queue definition object](#).

Use the TRIGDATA field on the transmission queue to trigger the specified channel. For example:

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

The supplied sample CSQ4INXD gives additional examples of the necessary definitions.

 Loss of connectivity to the CF structure where the synchronization queue for shared channels is defined, or similar problems, might temporarily prevent a channel from starting. After problem resolution, if you are using a trigger type of FIRST and the channel fails to start when it is triggered, you must start the channel manually. If you want to automatically start triggered channels after problem resolution, consider setting the queue manager TRIGINT attribute to a value other than the default. Setting the TRIGINT attribute to a value other than the default causes the channel initiator to retry starting the channel periodically while there are messages on the transmission queue.

### Synchronization queue

DQM requires a queue for use with sequence numbers and logical units of work identifiers (LUWID). You must ensure that a queue is available with the name SYSTEM.CHANNEL.SYNCQ (see [Pianificazione su z/OS](#)). This queue must be available otherwise the channel initiator cannot start.

Make sure that you define this queue using INDXTYPE(MSGID). This attribute improves the speed at which they can be accessed.

### Channel command queues

You need to ensure that a channel command queue exists for your system with the name SYSTEM.CHANNEL.INITQ.

If the channel initiator detects a problem with the SYSTEM.CHANNEL.INITQ, it is unable to continue normally until the problem is corrected. The problem could be one of the following:

- The queue is full
- The queue is not enabled for put
- The page set that the queue is on is full
- The channel initiator does not have the correct security authorization to the queue

If the definition of the queue is changed to GET(DISABLED) while the channel initiator is running, the initiator is unable to get messages from the queue, and terminates.

## Starting the channel initiator

Triggering is implemented using the channel initiator. On IBM MQ for z/OS, the initiator is started with the MQSC command `START CHINIT`.

## Stopping the channel initiator

The channel initiator is stopped automatically when you stop the queue manager. If you need to stop the channel initiator but not the queue manager, you can use the MQSC command `STOP CHINIT`.

## Monitoring and controlling channels on z/OS

Use the DQM commands and panels to create, monitor, and control the channels to remote queue managers.

Each z/OS queue manager has a DQM program (the *channel initiator*) for controlling interconnections to remote queue managers using native z/OS facilities.

The implementation of these panels and commands on z/OS is integrated into the operations and control panels and the MQSC commands. No differentiation is made in the organization of these two sets of panels and commands.

You can also enter commands using Programmable Command Format (PCF) commands. See [Automating administration tasks](#) for information about using these commands.

The information in this section applies in all cases where the channel initiator is used for distributed queuing. It applies whether you are using queue sharing groups, or intra-group queuing.

## The DQM channel control function

For an overview of the distributed queue management model, see [“Invio e ricezione di messaggi” on page 227](#).

The channel control function consists of panels, commands and programs, two synchronization queues, channel command queues, and the channel definitions. This topic is a brief description of the components of the channel control function.

- The channel definitions are held as objects in page set zero or in Db2, like other IBM MQ objects in z/OS.
- You use the operations and control panels, MQSC commands, or PCF commands to:
  - Create, copy, display, alter, and delete channel definitions
  - Start and stop channel initiators and listeners
  - Start, stop, and ping channels, reset channel sequence numbers, and resolve in-doubt messages when links cannot be re-established
  - Display status information about channels
  - Display information about DQM

In particular, you can use the CSQINPX initialization input data set to issue your MQSC commands. This set can be processed every time you start the channel initiator. For more information, see [Initialization commands](#).

- There are two queues (SYSTEM.CHANNEL.SYNCQ and SYSTEM.QSG.CHANNEL.SYNCQ) used for channel re-synchronization purposes. Define these queues with `INDXTYPE(MSGID)` for performance reasons.
- The channel command queue (SYSTEM.CHANNEL.INITQ) is used to hold commands for channel initiators, channels, and listeners.
- The channel control function program runs in its own address space, separate from the queue manager, and comprises the channel initiator, listeners, MCAs, trigger monitor, and command handler.

- For queue sharing groups and shared channels, see [Shared queues and queue sharing groups](#).
- For intra-group queuing, see [Intra-group queuing](#)

## Managing your channels on z/OS

Use the links in the following table for information about how to manage your channels, channel initiators, and listeners:

<i>Table 65. Channel tasks</i>	
<b>Task to be performed</b>	<b>MQSC command</b>
<a href="#">Define a channel</a>	<a href="#">DEFINE CHANNEL</a>
<a href="#">Alter a channel definition</a>	<a href="#">ALTER CHANNEL</a>
<a href="#">Display a channel definition</a>	<a href="#">DISPLAY CHANNEL</a>
<a href="#">Delete a channel definition</a>	<a href="#">DELETE CHANNEL</a>
<a href="#">Start a channel initiator</a>	<a href="#">START CHINIT</a>
<a href="#">Stop a channel initiator</a>	<a href="#">STOP CHINIT</a>
<a href="#">Display channel initiator information</a>	<a href="#">DISPLAY CHINIT</a>
<a href="#">Start a channel listener</a>	<a href="#">START LISTENER</a>
<a href="#">Stop a channel listener</a>	<a href="#">STOP LISTENER</a>
<a href="#">Start a channel</a>	<a href="#">START CHANNEL</a>
<a href="#">Test a channel</a>	<a href="#">PING CHANNEL</a>
<a href="#">Reset message sequence numbers for a channel</a>	<a href="#">RESET CHANNEL</a>
<a href="#">Resolve in-doubt messages on a channel</a>	<a href="#">RESOLVE CHANNEL</a>
<a href="#">Stop a channel</a>	<a href="#">STOP CHANNEL</a>
<a href="#">Display channel status</a>	<a href="#">DISPLAY CHSTATUS</a>
<a href="#">Display cluster channels</a>	<a href="#">DISPLAY CLUSQMGR</a>

### Related concepts

[“Using the panels and the commands” on page 967](#)

You can use the MQSC commands, the PCF commands, or the operations and control panels to manage DQM.

[“Setting up IBM MQ for z/OS” on page 888](#)

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

[“Setting up communication for z/OS” on page 981](#)

When a distributed-queuing management channel is started, it tries to use the connection specified in the channel definition. To succeed, it is necessary for the connection to be defined and available. This section explains how to define a connection.

[“Preparing IBM MQ for z/OS for DQM with queue sharing groups” on page 986](#)

Use the instructions in this section to configure distributed queuing with queue sharing groups on IBM MQ for z/OS.

[“Setting up communication for IBM MQ for z/OS using queue sharing groups” on page 990](#)

When a distributed-queuing management channel is started, it attempts to use the connection specified in the channel definition. For this attempt to succeed, it is necessary for the connection to be defined and available.

## Related tasks

“Configurazione dell'accodamento distribuito” on page 205

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

“Setting up communications with other queue managers on z/OS” on page 961

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

## Using the panels and the commands

You can use the MQSC commands, the PCF commands, or the operations and control panels to manage DQM.

For information about MQSC commands, see [Administering IBM MQ using MQSC commands](#). For information about PCF commands, see [Automating administration using Programmable Command Formats commands](#).

## Using the initial panel

For an introduction to invoking the operations and control panels, using the function keys, and getting help, see [Amministrazione di IBM MQ for z/OS](#).

**Note:** To use the operations and control panels, you must have the correct security authorization; see [Administering IBM MQ for z/OS](#) and sub topics for more information. [Figure 103 on page 967](#) shows the panel that is displayed when you start a panel session. The text after the panel explains the actions you perform in this panel.

```
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 MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time 10 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
```

*Figure 103. The operations and controls initial panel*

From this panel, you can:

- Select the action you want to perform by typing in the appropriate number in the **Action** field.
- Specify the object type that you want to work with. Press F4 for a list of object types if you are not sure what they are.
- Display a list of objects of the type specified. Type in an asterisk (\*) in the **Name** field and press enter to display a list of objects (of the type specified) that have already been defined on this subsystem. You

can then select one or more objects to work with in sequence. [Figure 104 on page 968](#) shows a list of channels produced in this way.

- Specify the disposition in the queue sharing group of the objects you want to work with in the **Disposition** field. The disposition determines where the object is kept and how the object behaves.
- Choose the local queue manager, or queue sharing group to which you want to connect in the **Connect name** field. If you want the commands to be issued on a remote queue manager, choose either the **Target queue manager** field or the **Action queue manager** field, depending upon whether the remote queue manager is not or is a member of a queue sharing group. If the remote queue manager is not a member of a queue sharing group, choose the **Target queue manager** field. If the remote queue manager is a member of a queue sharing group, choose the **Action queue manager** field.
- Choose the wait time for responses to be received in the **Response wait time** field.

```
List Channels - MQ25 Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.
1=Display 2=Define like 3=Alter 4=Manage 5=Perform
6=Start 7=Stop

Name Type Disposition Status
<> * CHANNEL ALL MQ25
- SYSTEM.DEF.CLNTCONN CLNTCONN QMGR MQ25
- SYSTEM.DEF.CLUSRCVR CLUSRCVR QMGR MQ25 INACTIVE
- SYSTEM.DEF.CLUSSDR CLUSSDR QMGR MQ25 INACTIVE
- SYSTEM.DEF.RECEIVER RECEIVER QMGR MQ25 INACTIVE
- SYSTEM.DEF.REQUESTER REQUESTER QMGR MQ25 INACTIVE
- SYSTEM.DEF.SENDER SENDER QMGR MQ25 INACTIVE
- SYSTEM.DEF.SERVER SERVER QMGR MQ25 INACTIVE
- SYSTEM.DEF.SVRCONN SVRCONN QMGR MQ25 INACTIVE
***** End of list *****

Command ==>> -----
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Status F12=Cancel
```

Figure 104. Listing channels

## Defining a channel on z/OS

On z/OS, you can define a channel by using MQSC commands or using the operations and control panels.

### Procedure

- To define a channel using the MQSC commands, use the **DEFINE CHANNEL** command.
- To use the operations and control panels, starting from the initial panel, complete the following fields and press Enter:

Table 66. Operations and control panels: initial panel fields	
Field	Value to enter in field
Action	2 (Define like)
Object type	Channel type (for example SENDER) or CHANNEL
Name	
Disposition	The location of the new object.

You are presented with some panels to complete with information about the name and attributes that you want for the channel you are defining. They are initialized with the default attribute values. Change any that you want to before pressing Enter.



**Note:** If you entered CHANNEL in the **object type** field, you are presented with the **Select a Valid Channel Type** panel first.

If you want to define a channel with the same attributes as an existing channel, put the name of the channel you want to copy in the **Name** field on the initial panel. The panels are initialized with the attributes of the existing object.

For information about the channel attributes, see [Channel attributes](#).

**Note:**

1. Name all the channels in your network uniquely. As shown in [Network diagram showing all channels](#), including the source and target queue manager names in the channel name is a good way to do this naming.

## What to do next

After you have defined your channel you must secure your channel. For more information, see [“Securing a channel”](#) on page 970.

### **Altering a channel definition**

You can alter a channel definition using MQSC commands or using the operations and control panels.

To alter a channel definition using the MQSC commands, use ALTER CHANNEL.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	3 (Alter)
Object type	channel type (for example SENDER) or CHANNEL
Name	CHANNEL.TO.ALTER
Disposition	The location of the stored object.

You are presented with some panels containing information about the current attributes of the channel. Change any of the unprotected fields that you want by over typing the new value, and then press enter to change the channel definition.

For information about the channel attributes, see [Channel attributes](#).

### **Displaying a channel definition**

You can display a channel definition using MQSC commands or using the operations and control panels.

To display a channel definition using the MQSC commands, use DISPLAY CHANNEL.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	1 (List or Display)
Object type	channel type (for example SENDER) or CHANNEL
Name	CHANNEL.TO.DISPLAY
Disposition	The location of the object.

You are presented with some panels displaying information about the current attributes of the channel.

For information about the channel attributes, see [Channel attributes](#).

## **Deleting a channel definition**

You can delete a channel definition using MQSC commands or using the operations and control panels.

To delete a channel definition using the MQSC commands, use `DELETE CHANNEL`.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

<b>Field</b>	<b>Value</b>
Action	4 (Manage)
Object type	channel type (for example SENDER) or CHANNEL
Name	CHANNEL.TO.DELETE
Disposition	The location of the object.

You are presented with another panel. Select function type 1 on this panel.

Press enter to delete the channel definition; you are asked to confirm that you want to delete the channel definition by pressing enter again.

**Note:** The channel initiator has to be running before a channel definition can be deleted (except for client-connection channels).

## **Displaying information about the channel initiator**

You can display information about the channel initiator using MQSC commands or using the operations and control panels.

To display information about the channel initiator using the MQSC commands, use `DISPLAY CHINIT`.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

<b>Field</b>	<b>Value</b>
Action	1 (Display)
Object type	SYSTEM
Name	Blank

You are presented with another panel. Select function type 1 on this panel.

**Note:**

1. Displaying distributed queuing information might take some time if you have lots of channels.
2. The channel initiator has to be running before you can display information about distributed queuing.

## **Securing a channel**

You can secure a channel using MQSC commands or using the operations and control panels.

To secure a channel using the MQSC commands, use `SET CHLAUTH`.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

<b>Field</b>	<b>Value</b>
Action	8

You are presented with an editor within which you can provide an MQSC command, in this case a `CHLAUTH` command, see [Figure 105 on page 971](#). When you have finished typing in the command, the

plus signs (+) are needed. Type PF3 to exit from the editor and submit the command to the command server.

```
***** Top of Data *****
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
***** Bottom of Data *****

Command ==> Scroll ==> PAGE
F1=Help F3=Exit F4=LineEdit F12=Cancel
```

Figure 105. Command Entry

The output of the command is then presented to you, see [Figure 106 on page 971](#)

```
***** Top of Data *****
000001 CSQU000I CSQUTIL IBM MQ for z/OS V7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMCA ' SET CHLAUTH' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ==> Scroll ==> PAGE
F1=Help F3=Exit F5=Rfind F6=Rchange F9=SwapNext F12=Cancel
```

Figure 106. Command Output

## Starting a channel initiator

You can start a channel initiator using MQSC commands or using the operations and control panels.

To start a channel initiator using the MQSC commands, use START CHINIT.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	6 (Start)
Object type	SYSTEM
Name	Blank

The Start a System Function panel is displayed. The text following the following panel explains what action to take:

### Start a System Function

Select function type, complete fields, then press Enter to start system function.

```
Function type _ 1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
JCL substitution -----

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type _ L=LU6.2, T=TCP/IP
LU name (LU6.2) -----
Port number (TCP/IP) . . . 1414
IP address (TCP/IP) . . . -----

Command ==> -----
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

Figure 107. Starting a system function

Select function type 1 (channel initiator), and press enter.

### Stopping a channel initiator

You can stop a channel initiator using MQSC commands or using the operations and control panels.

To stop a channel initiator using the MQSC commands, use STOP CHINIT.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	7 (Stop)
Object type	SYSTEM
Name	Blank

The Stop a System Function panel is displayed. The text following the panel explains how you to use this panel:

```

Stop a System Function

Select function type, complete fields, then press Enter to stop system
function.

Function type _ 1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
Restart shared channels Y Y=Yes, N=No

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type _ L=LU6.2, T=TCP/IP

Port number (TCP/IP) . . . -----
IP address (TCP/IP) . . . -----

Command ==> -----
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

```

Figure 108. Stopping a function control

Select function type 1 (channel initiator) and press enter.

The channel initiator waits for all running channels to stop in quiesce mode before it stops.

**Note:** If some of the channels are receiver or requester channels that are running but not active, a stop request issued to either the receiver or sender channel initiator causes it to stop immediately.

However, if messages are flowing, the channel initiator waits for the current batch of messages to complete before it stops.

### Starting a channel listener

You can start a channel listener using MQSC commands or using the operations and control panels.

To start a channel listener using the MQSC commands, use START LISTENER.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	6 (Start)
Object type	SYSTEM
Name	Blank

The Start a System Function panel is displayed (see [Figure 107 on page 972](#) ).

Select function type 2 (channel listener). Select Inbound disposition. Select Transport type. If the Transport type is L, select LU name. If the Transport type is T, select Port number and (optionally) IP address. Press enter.

**Note:** For the TCP/IP listener, you can start multiple combinations of Port and IP address.

### Stopping a channel listener

You can stop a channel listener using MQSC commands or using the operations and control panels.

To stop a channel listener using the MQSC commands, use STOP LISTENER.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	7 (Stop)
Object type	SYSTEM
Name	Blank

The Stop a System Function panel is displayed (see [Figure 108 on page 973](#)).

Select function type 2 (channel listener). Select Inbound disposition. Select Transport type. If the transport type is 'T', select Port number and (optionally) IP address. Press enter.

**Note:** For a TCP/IP listener, you can stop specific combinations of Port and IP address, or you can stop all combinations.

### **Starting a channel**

You can start a channel using MQSC commands or using the operations and control panels.

To start a channel using the MQSC commands, use START CHANNEL.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	6 (Start)
Object type	channel type (for example SENDER) or CHANNEL
Name	CHANNEL.TO.USE
Disposition	The disposition of the object.

The Start a Channel panel is displayed. The text following the panel explains how to use the panel:

```

Start a Channel
Select disposition, then press Enter to start channel.

Channel name : CHANNEL.TO.USE
Channel type : SENDER
Description : Description of CHANNEL.TO.USE

Disposition P P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Command ==> _____
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

```

*Figure 109. Starting a channel*

Select the disposition of the channel instance and on which queue manager it is to be started.

Press enter to start the channel.

## **Starting a shared channel**

To start a shared channel, and keep it on a nominated channel initiator, use disposition = S (on the START CHANNEL command, specify CHLDISP(FIXSHARED)).

There can be only one instance of the shared channel running at a time. Attempts to start a second instance of the channel fail.

When you start a channel in this way, the following rules apply to that channel:

- You can stop the channel from any queue manager in the queue sharing group. You can stop it even if the channel initiator on which it was started is not running at the time you issue the stop-channel request. When the channel has stopped, you can restart it by specifying disposition = S (CHLDISP(FIXSHARED)) on the same, or another, channel initiator. You can also start it by specifying disposition = A (CHLDISP(SHARED)).
- If the channel is in the starting or retry state, you can restart it by specifying disposition = S (CHLDISP(FIXSHARED)) on the same or a different channel initiator. You can also start it by specifying disposition = A (CHLDISP(SHARED)).
- The channel is eligible to be trigger started when it goes into the inactive state. Shared channels that are trigger started always have a shared disposition (CHLDISP(SHARED)).
- The channel is eligible to be started with CHLDISP(FIXSHARED), on any channel initiator, when it goes into the inactive state. You can also start it by specifying disposition = A (CHLDISP(SHARED)).
- The channel is not recovered by any other active channel initiator in the queue sharing group when the channel initiator on which it was started is stopped with SHARED(RESTART), or when the channel initiator terminates abnormally. The channel is recovered only when the channel initiator on which it was started is next restarted. This stops failed channel-recovery attempts being passed to other channel initiators in the queue sharing group, which would add to their workload.

## **Testing a channel**

You can test a channel using MQSC commands or using the operations and control panels.

To test a channel using the MQSC commands, use PING CHANNEL.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

<b>Field</b>	<b>Value</b>
Action	5 (Perform)
Object type	SENDER, SERVER, or CHANNEL
Name	CHANNEL.TO.USE
Disposition	The disposition of the channel object.

The Perform a Channel Function panel is displayed. The text following the panel explains how to use the panel:

### Perform a Channel Function

Select function type, complete fields, then press Enter.

```
Function type _ 1. Reset 3. Resolve with commit
2. Ping 4. Resolve with backout
```

```
Channel name : CHANNEL.TO.USE
Channel type : SENDER
Description : Description of CHANNEL.TO.USE
```

```
Disposition P P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager
```

```
Sequence number for reset . . 1 1 - 99999999
Data length for ping 16 16 - 32768
```

```
Command ==> _____
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

Figure 110. Testing a channel

Select function type 2 (ping).

Select the disposition of the channel for which the test is to be done and on which queue manager it is to be tested.

The data length is initially set to 16. Change it if you want and press enter.

### **Resetting message sequence numbers for a channel**

You can reset message sequence numbers for a channel using MQSC commands or using the operations and control panels.

To reset channel sequence numbers using the MQSC commands, use RESET CHANNEL.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	5 (Perform)
Object type	channel type (for example SENDER) or CHANNEL
Name	CHANNEL.TO.USE
Disposition	The disposition of the channel object.

The Perform a Channel Function panel is displayed (see [Figure 110 on page 976](#) ).

Select Function type 1 (reset).

Select the disposition of the channel for which the reset is to be done and on which queue manager it is to be done.

The **sequence number** field is initially set to one. Change this value if you want, and press enter.

### **Resolving in-doubt messages on a channel**

You can resolve in-doubt messages on a channel using MQSC commands or using the operations and control panels.

To resolve in-doubt messages on a channel using the MQSC commands, use RESOLVE CHANNEL.



Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	5 (Perform)
Object type	SENDER, SERVER, or CHANNEL
Name	CHANNEL.TO.USE
Disposition	The disposition of the object.

The Perform a Channel Function panel is displayed (see [Figure 110 on page 976](#)).

Select Function type 3 or 4 (resolve with commit or backout). (See [“Gestione dei canali in dubbio” on page 247](#) for more information.)

Select the disposition of the channel for which resolution is to be done and which queue manager it is to be done on. Press enter.

### **Stopping a channel**

You can stop a channel using MQSC commands or using the operations and control panels.

To stop a channel using the MQSC commands, use STOP CHANNEL.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	7 (Stop)
Object type	channel type (for example SENDER) or CHANNEL
Name	CHANNEL.TO.USE
Disposition	The disposition of the object.

The Stop a Channel panel is displayed. The text following the panel explains how to use the panel:

```

Stop a Channel
Complete fields, then press Enter to stop channel.

Channel name : CHANNEL.TO.USE
Channel type : SENDER
Description : Description of CHANNEL.TO.USE

Disposition P P=Private on MQ25
A=Shared on any queue manager

Stop mode 1 1. Quiesce 2. Force
Stop status 1 1. Stopped 2. Inactive

Queue manager -----
Connection name -----

Command ==> -----
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

```

*Figure 111. Stopping a channel*

Select the disposition of the channel for which the stop is to be done and on which queue manager it is to be stopped.

Choose the stop mode that you require:

### Quiesce

The channel stops when the current message is completed and the batch is then ended, even if the batch size value has not been reached and there are messages already waiting on the transmission queue. No new batches are started. This mode is the default.

### Force

The channel stops immediately. If a batch of messages is in progress, an 'in-doubt' situation can result.

Choose the queue manager and connection name for the channel you want to stop.

Choose the status that you require:

### Stopped

The channel is not restarted automatically, and must be restarted manually. This mode is the default if no queue manager or connection name is specified. If a name is specified, it is not allowed.

### Inactive

The channel is restarted automatically when required. This mode is the default if a queue manager or connection name is specified.

Press enter to stop the channel.

See [“Arresto e disattivazione dei canali” on page 245](#) for more information. For information about restarting stopped channels, see [“Riavvio dei canali arrestati” on page 246](#).

**Note:** If a shared channel is in a retry state and the channel initiator on which it was started is not running, a STOP request for the channel is issued on the queue manager where the command was entered.

## **Displaying channel status**

You can display channel status by using MQSC commands, or by using the operations and control panels.

To display the status of a channel or a set of channels using the MQSC commands, use DISPLAY CHSTATUS.

**Note:** Displaying channel status information can take some time if you have lots of channels.

Using the operations and control panels on the List Channel panel (see [Figure 104 on page 968](#)), a summary of the channel status is shown for each channel as follows:

INACTIVE	No connections are active
<i>status</i>	One connection is active
<i>nnn status</i>	More than one connection is current and all current connections have the same status
<i>nnn CURRENT</i>	More than one connection is current and the current connections do not all have the same status
Blank	IBM MQ is unable to determine how many connections are active (for example, because the channel initiator is not running)

**Note:** For channel objects with the disposition GROUP, no status is displayed.

where *nnn* is the number of active connections, and *status* is one of the following:

INIT	INITIALIZING
BIND	BINDING
START	STARTING

RUN	RUNNING
STOP	STOPPING or STOPPED
RETRY	RETRYING
REQST	REQUESTING

To display more information about the channel status, press the Status key (F11) on the List Channel or the Display, or Alter channel panels to display the List Channels - Current Status panel (see [Figure 112](#) on page 979).

```

List Channels - Current Status - MQ25 Row 1 of 16

Type action codes, then press Enter. Press F11 to display saved status.
1=Display current status

Channel name Connection name State
Start time Messages Last message time Type Disposition
<> *
CHANNEL ALL MQ25

- RMA0.CIRCUIT.ACL.F RMA1 STOP
- 2005-03-21 10.22.36 557735 2005-03-24 09.51.11 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.ACL.N RMA1
- 2005-03-21 10.23.09 378675 2005-03-24 09.51.10 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.CL.F RMA2
- 2005-03-24 01.12.51 45544 2005-03-24 09.51.08 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.CL.N RMA2
- 2005-03-24 01.13.55 45560 2005-03-24 09.51.11 SENDER PRIVATE MQ25
- RMA1.CIRCUIT.CL.F RMA1
- 2005-03-21 10.24.12 360757 2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
- RMA1.CIRCUIT.CL.N RMA1
- 2005-03-21 10.23.40 302870 2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
***** End of list *****
Command ==>
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Saved F12=Cancel

```

Figure 112. Listing channel connections

The values for status are as follows:

INIT	INITIALIZING
BIND	BINDING
START	STARTING
RUN	RUNNING
STOP	STOPPING or STOPPED
RETRY	RETRYING
REQST	REQUESTING
DOUBT	STOPPED and INDOUBT(YES)

See [“Stati del canale”](#) on page 237 for more information.

You can press F11 to see a similar list of channel connections with saved status; press F11 to get back to the current list. The saved status does not apply until at least one batch of messages has been transmitted on the channel.

Use action code 1 or a slash (/) to select a connection and press enter. The Display Channel Connection Current Status panels are displayed.

## **Displaying cluster channels**

You can display cluster channels using MQSC commands or using the operations and control panels.

To display all the cluster channels that have been defined (explicitly or using auto-definition), use the MQSC command, DISPLAY CLUSQMGR.

Using the operations and control panels, starting from the initial panel, complete these fields and press enter:

Field	Value
Action	1 (List or Display)
Object type	CLUSCHL
Name	*

You are presented with a panel like figure [Figure 113 on page 980](#), in which the information for each cluster channel occupies three lines, and includes its channel, cluster, and queue manager names. For cluster-sender channels, the overall state is shown.

```
List Cluster queue manager Channels - MQ25 Row 1 of 9

Type action codes, then press Enter. Press F11 to display connection status.
1=Display 5=Perform 6=Start 7=Stop

Channel name Connection name State
Type Cluster name Suspended
Cluster queue manager name Disposition
<> * - MQ25
- TO.MQ90.T HURSLEY.MACH90.COM(1590)
- CLUSRCVR VJH01T N
- MQ90 - MQ25
- TO.MQ95.T HURSLEY.MACH95.COM(1595) RUN
- CLUSSDRA VJH01T N
- MQ95 - MQ25
- TO.MQ96.T HURSLEY.MACH96.COM(1596) RUN
- CLUSSDRB VJH01T N
- MQ96 - MQ25
***** End of list *****

Command ==>-----
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Status F12=Cancel
```

Figure 113. Listing cluster channels

To display full information about one or more channels, type Action code 1 against their names and press enter. Use Action codes 5, 6, or 7 to perform functions (such as ping, resolve, and reset), and start or stop a cluster channel.

To display more information about the channel status, press the Status key (F11).

## **Preparing IBM MQ for z/OS to use the zEnterprise Data Compression**

### **Express facility**

The zEnterprise® Data Compression (zEDC) Express facility is available for certain models of IBM Z machines, starting from IBM zEC12 GA2, using a minimum z/OS level of z/OS 2.1.

See [zEnterprise Data Compression \(zEDC\)](#) for further information.

### **Prerequisites**

For IBM z15 and later, the zEnterprise Data Compression (zEDC) Express facility was moved from an optional feature in the PCIe I/O drawer of the hardware system to be on-chip as the Integrated

Accelerator for zEDC. With this change, the configuration prerequisites are updated and are dependent on your hardware system.

### **IBM z15 or later**

Apply one of the following PTFs, according to your level of z/OS:

- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638
- z/OS 2.1: UJ00639

There are no hardware requirements for z15 or later systems. The Integrated Accelerator for zEDC solution in these systems provides built-in data acceleration, so a separate adapter is no longer required.

### **IBM zEC12 GA2 to IBM z14**

Your system must also have the following requirements:

- A zEDC Express<sup>®</sup> adapter, installed in the PCIe I/O drawers of the hardware system.
- The zEDC software capability (an optional, paid-for feature) must be enabled in an IFAPRDxx parmlib member.

## **Procedure**

### **IBM zEC12 GA2 to IBM z14**

Ensure that the channel initiator user ID has READ authority to the FPZ.ACCELERATOR.COMPRESSION profile in the RACF FACILITY CLASS, or the equivalent in the external security manager (ESM) that your enterprise uses.



**Attention:** Not required for IBM z15 or later.

### **IBM zEnterprise zEC12 GA2 or later**

Configure the channel with COMPMSG(ZLIBFAST) at both the sending and receiving ends. Once configured, zlib compression is used to compress and decompress messages flowing across the channel.

Compression is performed in the zEDC when the size of the data to be compressed is above the minimum threshold. The threshold is dependent upon the IBM z hardware being used

- IBM zEC12 GA2 to IBM z14 has a minimum threshold of 4KB
- IBM z15 or later has a minimum threshold of 1KB

For messages below the threshold size, compression or inflation is performed in the software.

## **z/OS Setting up communication for z/OS**

When a distributed-queuing management channel is started, it tries to use the connection specified in the channel definition. To succeed, it is necessary for the connection to be defined and available. This section explains how to define a connection.

DQM is a remote queuing facility for IBM MQ. It provides channel control programs for the queue manager that form the interface to communication links. These links are controllable by the system operator. The channel definitions held by distributed queuing management use these connections.

Choose from one of the two forms of communication protocol that can be used for z/OS:

- [“Defining a TCP connection on z/OS” on page 982](#)
- [“Defining an LU6.2 connection for z/OS using APPC/MVS” on page 985](#)

Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linux o Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

Each channel definition must specify only one protocol as the transmission protocol (Transport Type) attribute. A queue manager can use more than one protocol to communicate.

You might also find it helpful to refer to [Example configuration - IBM MQ for z/OS](#). If you are using queue sharing groups, see [“Setting up communication for IBM MQ for z/OS using queue sharing groups” on page 990](#).

### Related concepts

[“Using the panels and the commands” on page 967](#)

You can use the MQSC commands, the PCF commands, or the operations and control panels to manage DQM.

[“Setting up IBM MQ for z/OS” on page 888](#)

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

[“Monitoring and controlling channels on z/OS” on page 965](#)

Use the DQM commands and panels to create, monitor, and control the channels to remote queue managers.

[“Preparing IBM MQ for z/OS for DQM with queue sharing groups” on page 986](#)

Use the instructions in this section to configure distributed queuing with queue sharing groups on IBM MQ for z/OS.

[“Setting up communication for IBM MQ for z/OS using queue sharing groups” on page 990](#)

When a distributed-queuing management channel is started, it attempts to use the connection specified in the channel definition. For this attempt to succeed, it is necessary for the connection to be defined and available.

### Related tasks

[“Configurazione dell'accodamento distribuito” on page 205](#)

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

[“Setting up communications with other queue managers on z/OS” on page 961](#)

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

## **z/OS** *Defining a TCP connection on z/OS*

To define a TCP connection, there are a number of settings to configure.

The TCP address space name must be specified in the TCP system parameters data set, *tcPIP.TCPIP.DATA*. In the data set, a "TCPIPJOBNAME *TCPIP\_proc*" statement must be included.

If you are using a firewall, you need to configure allow connections from the channel initiator to the addresses in the channels, and from the remote connections into the queue manager.

Typically the definition for a firewall configures the sending IP address and port to the destination IP address and port:

- A z/OS image can have more than one host name, and you might need to configure the firewall with multiple host addresses as the source address.

You can use the NETSTAT HOME command to display these names and addresses.

- A channel initiator can have multiple listeners on different ports, so you need to configure these ports.
- If you are using a shared port for a queue sharing group you must configure the shared port as well.

The channel initiator address space must have authority to read the data set. The following techniques can be used to access your TCPIP.DATA data set, depending on which TCP/IP product and interface you are using:

- Environment variable, RESOLVER\_CONFIG
- /etc/resolv.conf on the file system
- //SYSTCPD DD statement
- //SYSTCPDD DD statement
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

You must also be careful to specify the high-level qualifier for TCP/IP correctly.

You need a suitably configured Domain Name System (DNS) server, capable of both Name to IP address translation and IP address to Name translation.

**Note:** Some changes to the resolver configuration require a recycle of applications using it, for example, IBM MQ.

For more information, see the following:

- [Base TCP/IP system](#)
- [z/OS UNIX System Services](#).

Each TCP channel when started uses TCP resources; you might need to adjust the following parameters in your PROFILE.TCPIP configuration data set:

#### **ACBPOOLSIZE**

Add one per started TCP channel, plus one

#### **CCBPOOLSIZE**

Add one per started TCP channel, plus one per DQM dispatcher, plus one

#### **DATABUFFERPOOLSIZE**


Add two per started TCP channel, plus one

#### **MAXFILEPROC**

Controls how many channels each dispatcher in the channel initiator can handle.

This parameter is specified in the BPXPRMxx member of SYSI.PARMLIB. Ensure that you specify a value large enough for your needs.

By default, the channel initiator is only capable of binding to IP addresses associated with the stack named in the TCPNAME queue manager attribute. To allow the channel initiator to communicate using additional TCP/IP stacks on the system, change the TCPSTACK queue manager attribute to MULTIPLE.

 Un canale di messaggi che utilizza TCP/IP può essere puntato a un IBM Aspera faspio Gateway, che fornisce un tunnel TCP/IP veloce che può aumentare significativamente la velocità di trasmissione di rete. Un gestore code in esecuzione su qualsiasi piattaforma autorizzata può connettersi tramite un Aspera gateway. Il gateway stesso è distribuito su Red Hat o Ubuntu Linuxo Windows. Consultare [Definizione di una connessione Aspera gateway su Linux o Windows](#).

#### **Related concepts**

[“Sending end” on page 984](#)

At the sending end of the TCP/IP connection, there are a number of settings to configure.

[“Receiving on TCP” on page 984](#)

At the receiving end of the TCP/IP connection, there are a number of settings to configure.

[“Using the TCP listener backlog option on z/OS” on page 984](#)

When receiving on TCP/IP, a maximum number of outstanding connection requests is set. These outstanding requests can be considered a *backlog* of requests waiting on the TCP/IP port for the listener to accept the request.

## *Sending end*

At the sending end of the TCP/IP connection, there are a number of settings to configure.

The connection name (CONNNAME) field in the channel definition must be set to either the host name (for example MVSHUR1) or the TCP network address of the target. The TCP network address can be in IPv4 dotted decimal form (for example 127.0.0.1) or IPv6 hexadecimal form (for example 2001:DB8:0:0:0:0:0:0). If the connection name is a host name, a TCP name server is required to convert the host name into a TCP host address. (This requirement is a function of TCP, not IBM MQ.)

On the initiating end of a connection (sender, requester, and server channel types) it is possible to provide an optional port number for the connection, for example:

**Connection name**  
192.0.2.0(1555)

In this case the initiating end attempts to connect to a receiving program listening on port 1555.

**Note:** The default port number of 1414 is used if an optional port number is not specified.

The channel initiator can use any TCP/IP stack which is active and available. By default, the channel initiator binds its outbound channels to the default IP address for the TCP/IP stack named in the TCPNAME queue manager attribute. To connect through a different stack, you need to specify either the host name or IP address of the stack in the LOCLADDR attribute of the channel.

## *Receiving on TCP*

At the receiving end of the TCP/IP connection, there are a number of settings to configure.

Receiving channel programs are started in response to a startup request from the sending channel. To do so, a listener program has to be started to detect incoming network requests and start the associated channel. You start this listener program with the [START LISTENER](#) command, or using the operations and control panels.

By default:

- The TCP Listener program uses port 1414 and listens on all addresses available to your TCP stack.
- TCP/IP listeners can bind only to addresses associated with the TCP/IP stack named in the TCPNAME queue manager attribute.

To start listeners for other addresses, or all available TCP stacks, set your TCPSTACK queue manager attribute to 'MULTIPLE'.

You can start your TCP listener program to listen only on a specific address or host name by specifying IPADDR in the START LISTENER command. For more information, see [Listeners](#).

## *Using the TCP listener backlog option on z/OS*

When receiving on TCP/IP, a maximum number of outstanding connection requests is set. These outstanding requests can be considered a *backlog* of requests waiting on the TCP/IP port for the listener to accept the request.

The default listener backlog value on z/OS is 10000. If the backlog reaches this values, the TCP/IP connection is rejected and the channel is not able to start.

For MCA channels, this results in the channel going into a RETRY state and retrying the connection at a later time.

For client connections, the client receives an MQRC\_Q\_MGR\_NOT\_AVAILABLE reason code from MQCONN and can retry the connection at a later time.

### **Related concepts**

[“Utilizzo dell'opzione di backlog del listener TCP su IBM MQ for Multiplatforms” on page 279](#)

In TCP, le connessioni sono considerate incomplete a meno che non si verifichi un handshake a tre vie tra il server e il client. Queste connessioni vengono chiamate richieste di connessione in sospenso. Viene impostato un valore massimo per queste richieste di connessione in sospenso e può essere considerato un backlog di richieste in attesa sulla porta TCP affinché il listener accetti la richiesta.



## Defining an LU6.2 connection for z/OS using APPC/MVS

To define an LU6.2 connection there are a number of settings to configure.

### APPC/MVS setup

Each instance of the channel initiator must have the name of the LU that it is to use defined to APPC/MVS, in the APPCPMxx member of SYS1.PARMLIB, as in the following example:

```
LUADD ACBNAME(luname) NOSCHED TPDATA(CSQ.APPCTP)
```

*luname* is the name of the logical unit to be used. NOSCHED is required; TPDATA is not used. No additions are necessary to the ASCHPMxx member, or to the APPC/MVS TP profile data set.

The side information data set must be extended to define the connections used by DQM. See the supplied sample CSQ4SIDE for details of how to do this using the APPC utility program ATBSDFMU. For details of the TPNAME values to use, see the following table for information:

Remote platform	TPNAME
z/OS or MVS	The same as TPNAME in the corresponding side information about the remote queue manager.
IBM i	The same as the compare value in the routing entry on the IBM i system.
AIX and Linux systems	The same as TPNAME in the corresponding side information about the remote queue manager.
Windows	As specified in the Windows Run Listener command, or the invokable Transaction Program that was defined using TpSetup on Windows.

If you have more than one queue manager on the same machine, ensure that the TPnames in the channel definitions are unique.

In an environment where the queue manager is communicating using APPC with a queue manager on the same or another z/OS system, ensure that either the VTAM definition for the communicating LU specifies SECACPT(ALREADYV), or that there is a RACF APPCLU profile for the connection between LUs, which specifies CONVSEC(ALREADYV).

The z/OS command VARY ACTIVE must be issued against both base and listener LUs before attempting to start either inbound or outbound communications.



**Attention:** In addition to the APPC setup, you must issue the following command:

```
ALTER QMGR LUNAME(luname)
```

and restart the channel initiator.

See [LUNAME](#) for further information.

### Related concepts

[“Connecting to LU 6.2” on page 985](#)

To connect to LU 6.2, there are a number of settings to configure.

[“Receiving on LU 6.2” on page 986](#)

To receive on LU 6.2, there are a number of settings to configure.



### Connecting to LU 6.2

To connect to LU 6.2, there are a number of settings to configure.

The connection name (CONNAME) field in the channel definition must be set to the symbolic destination name, as specified in the side information data set for APPC/MVS.

The LU name to use (defined to APPC/MVS as described previously) must also be specified in the channel initiator parameters. It must be set to the same LU that is used for receiving by the listener.

The channel initiator uses the "SECURITY(SAME)" APPC/MVS option, so it is the user ID of the channel initiator address space that is used for outbound transmissions, and is presented to the receiver.

### Receiving on LU 6.2

To receive on LU 6.2, there are a number of settings to configure.

Receiving MCAs are started in response to a startup request from the sending channel. To do so, a listener program has to be started to detect incoming network requests and start the associated channel. The listener program is an APPC/MVS server. You start it with the START LISTENER command, or using the operations and control panels. You must specify the LU name to use with a symbolic destination name defined in the side information data set. The local LU so identified must be the same as the one used for outbound transmissions, as set in the channel initiator parameters.

### Preparing IBM MQ for z/OS for DQM with queue sharing groups

Use the instructions in this section to configure distributed queuing with queue sharing groups on IBM MQ for z/OS.

For an example configuration using queue sharing groups, see [Example configuration - IBM MQ for z/OS using queue sharing groups](#). For a message channel planning example using queue sharing groups, see [Message channel planning example for z/OS using queue sharing groups](#).

You need to create and configure the following components to enable distributed queuing with queue sharing groups:

- [LU 6.2 and TCP/IP listeners](#)
- [Transmission queues and triggering](#)
- [Message channel agents](#)
- [Synchronization queue](#)

After you have created the components you need to set up the communication, see [“Setting up communication for IBM MQ for z/OS using queue sharing groups”](#) on page 990.

For information about how to monitor and control channels when using queue sharing groups, see [“Monitoring and controlling channels on z/OS”](#) on page 965.

See the following sections for queue sharing group concepts and benefits.

## Class of service

A shared queue is a type of local queue that offers a different class of service. Messages on a shared queue are stored in a coupling facility (CF), which allows them to be accessed by all queue managers in the queue sharing group. A message on a shared queue must be a message of length no more than 100 MB.

## Generic interface

A queue sharing group has a generic interface that allows the network to view the group as a single entity. This view is achieved by having a single generic address that can be used to connect to any queue manager within the group.

Each queue manager in the queue sharing group listens for inbound session requests on an address that is logically related to the generic address. For more information see [“LU 6.2 and TCP/IP listeners for queue sharing groups”](#) on page 988.

## Load-balanced channel start

A shared transmission queue can be serviced by an outbound channel running on any channel initiator in the queue sharing group. Load-balanced channel start determines where a start channel command is targeted. An appropriate channel initiator is chosen that has access to the necessary communications subsystem. For example, a channel defined with TRPTYPE(LU6.2) cannot be started on a channel initiator that only has access to a TCP/IP subsystem.

The choice of channel initiator is dependent on the channel load and the headroom of the channel initiator. The channel load is the number of active channels as a percentage of the maximum number of active channels allowed as defined in the channel initiator parameters. The headroom is the difference between the number of active channels and the maximum number allowed.

Inbound shared channels can be load-balanced across the queue sharing group by use of a generic address, as described in [“LU 6.2 and TCP/IP listeners for queue sharing groups”](#) on page 988.

## Shared channel recovery

The following table shows the types of shared-channel failure and how each type is handled.

Type of failure:	What happens:
Channel initiator communications subsystem failure	The channels dependent on the communications subsystem enter channel retry, and are restarted on an appropriate queue sharing group channel initiator by a load-balanced start command.
Channel initiator failure	The channel initiator fails, but the associated queue manager remains active. The queue manager monitors the failure and initiates recovery processing.
Queue manager failure	The queue manager fails (failing the associated channel initiator). Other queue managers in the queue sharing group monitor the event and initiate peer recovery.
Shared status failure	Channel state information is stored in Db2, so a loss of connectivity to Db2 becomes a failure when a channel state change occurs. Running channels can carry on running without access to these resources. On a failed access to Db2, the channel enters retry.

Shared channel recovery processing on behalf of a failed system requires connectivity to Db2 to be available on the system managing the recovery to retrieve the shared channel status.

## Client channels

Client connection channels can benefit from the high availability of messages in queue sharing groups that are connected to the generic interface instead of being connected to a specific queue manager. For more information, see [Client connection channels](#).

### Related concepts

[Shared queues and queue sharing groups](#)

[“Setting up IBM MQ for z/OS”](#) on page 888

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

[“Clusters and queue sharing groups”](#) on page 989

You can make your shared queue available to a cluster in a single definition. To do so you specify the name of the cluster when you define the shared queue.

[“Channels and serialization”](#) on page 990

During shared queue peer recovery, message channel agents that process messages on shared queues serialize their access to the queues.

[Intra-group queuing](#)

## Related tasks

“Configurazione dell'accodamento distribuito” on page 205

Questa sezione fornisce informazioni più dettagliate sull'intercomunicazione tra installazioni IBM MQ , incluse la definizione della coda, la definizione del canale, il trigger e le procedure del punto di sincronizzazione

“Setting up communications with other queue managers on z/OS” on page 961

This section describes the IBM MQ for z/OS preparations you need to make before you can start to use distributed queuing.

### **LU 6.2 and TCP/IP listeners for queue sharing groups**

The group LU 6.2 and TCP/IP listeners listen on an address that is logically connected to the generic address.

For the LU 6.2 listener, the specified LUGROUP is mapped to the VTAM generic resource associated with the queue sharing group. For an example of setting up this technology, see [“Defining an LU6.2 connection for z/OS using APPC/MVS”](#) on page 985.

For the TCP/IP listener, the specified port can be connected to the generic address in one of the following ways:

- For a front-end router such as the IBM Network Dispatcher, inbound connect requests are forwarded from the router to the members of the queue sharing group.
- For TCP/IP Sysplex Distributor, each listener that is running and is listening on a particular address that is set up as a Distributed DVIPA is allocated a proportion of the incoming requests. For an example of setting up this technology, see [Using Sysplex Distributor](#)

### **Transmission queues and triggering for queue sharing groups**

A shared transmission queue is used to store messages before they are moved from the queue sharing group to the destination.

It is a shared queue and it is accessible to all queue managers in the queue sharing group.

## Triggering

A triggered shared queue can generate more than one trigger message for a satisfied trigger condition. There is one trigger message generated for each local initiation queue defined on a queue manager in the queue sharing group associated with the triggered shared queue.

For distributed queuing, each channel initiator receives a trigger message for a satisfied shared transmission queue trigger condition. However, only one channel initiator actually processes the triggered start, and the others fail safely. The triggered channel is then started with a load balanced start (see [“Preparing IBM MQ for z/OS for DQM with queue sharing groups”](#) on page 986 ) that is triggered to start channel QSG . TO . QM2. To create a shared transmission queue, use the IBM MQ commands (MQSC) as shown in the following example:

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

**Note:** If a shared queue is setup for triggering and connection to the Coupling Facility hosting the shared queue is lost, a trigger event might be generated and a message put to the initiation queue. This can happen even when no message was put to the original shared queue setup for triggering. This is caused by the over-indication of bits by the IXLVECTR macro as documented in [The List Notification Vector](#).

### **Message channel agents for queue sharing groups**

A channel can only be started on a channel initiator if it has access to a channel definition for a channel with that name.

A message channel agent is an IBM MQ program that controls the sending and receiving of messages. Message channel agents move messages from one queue manager to another; there is one message channel agent at each end of a channel.

A channel definition can be defined to be private to a queue manager or stored on the shared repository and available anywhere (a group definition). This means that a group defined channel is available on any channel initiator in the queue sharing group.

**Note:** The private copy of the group definition can be changed or deleted.

To create group channel definitions, use the IBM MQ commands (MQSC) as shown in the following examples:

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

There are two perspectives from which to look at the message channel agents used for distributed queuing with queue sharing groups:

## Inbound

An inbound channel is a shared channel if it is connected to the queue manager through the group listener. It is connected either through the generic interface to the queue sharing group, then directed to a queue manager within the group, or targeted at the group port of a specific queue manager or the luname used by the group listener.

## Outbound

An outbound channel is a shared channel if it moves messages from a shared transmission queue. In the example commands, sender channel QSG.TO.QM2 is a shared channel because its transmission queue, QM2 is defined with QSGDISP(SHARED).

### **Synchronization queue for queue sharing groups**

Shared channels have their own shared synchronization queue called SYSTEM.QSG.CHANNEL.SYNCQ.

This synchronization queue is accessible to any member of the queue sharing group. (Private channels continue to use the private synchronization queue. See [“Defining IBM MQ objects on z/OS” on page 964](#) ). This means that the channel can be restarted on a different queue manager and channel initiator instance within the queue sharing group in the event of failure of the communications subsystem, channel initiator, or queue manager. For further information, see [“Preparing IBM MQ for z/OS for DQM with queue sharing groups” on page 986](#).

DQM with queue sharing groups requires that a shared queue is available with the name SYSTEM.QSG.CHANNEL.SYNCQ. This queue must be available so that a group listener can successfully start.

If a group listener fails because the queue was not available, the queue can be defined and the listener can be restarted without recycling the channel initiator. The non-shared channels are not affected.

Make sure that you define this queue using INDXTYPE(MSGID). This definition improves the speed at which messages on the queue can be accessed.

### **Clusters and queue sharing groups**

You can make your shared queue available to a cluster in a single definition. To do so you specify the name of the cluster when you define the shared queue.

Users in the network see the shared queue as being hosted by each queue manager within the queue sharing group. (The shared queue is not advertised as being hosted by the queue sharing group). Clients can start sessions with all members of the queue sharing group to put messages to the same shared queue.

For more information, see [“Configurazione di un cluster di gestore code” on page 303.](#)

### **z/OS Channels and serialization**

During shared queue peer recovery, message channel agents that process messages on shared queues serialize their access to the queues.

If a queue manager in a queue sharing group fails while a message channel agent is dealing with uncommitted messages on one or more shared queues, the channel and the associated channel initiator will end, and shared queue peer recovery will take place for the queue manager.

Because shared queue peer recovery is an asynchronous activity, peer channel recovery might try to simultaneously restart the channel in another part of the queue sharing group before shared queue peer recovery is complete. If this event happens, committed messages might be processed ahead of the messages still being recovered. To ensure that messages are not processed out of sequence in this way, message channel agents that process messages on shared queues serialize their access to these queues.

An attempt to start a channel for which shared queue peer recovery is still in progress might result in a failure. An error message indicating that recovery is in progress is issued, and the channel is put into retry state. Once queue manager peer recovery is complete, the channel can restart at the time of the next retry.

An attempt to RESOLVE, PING, or DELETE a channel can fail for the same reason.

### **z/OS Setting up communication for IBM MQ for z/OS using queue sharing groups**

When a distributed-queuing management channel is started, it attempts to use the connection specified in the channel definition. For this attempt to succeed, it is necessary for the connection to be defined and available.

Choose from one of the two forms of communication protocol that can be used:

- [TCP](#)
- [LU 6.2 through APPC/MVS](#)

You might find it useful to refer to [Example configuration - IBM MQ for z/OS using queue sharing groups.](#)

### **z/OS Defining a TCP connection for queue sharing groups**

To define a TCP connection for a queue sharing group, certain attributes on the sending and receiving end must be configured.

For information about setting up your TCP, see [“Defining a TCP connection on z/OS” on page 982.](#)

## **Sending end**

The connection name (CONNNAME) field in the channel definition to connect to your queue sharing group must be set to the generic interface of your queue sharing group (see [Queue sharing groups](#)). For more details, refer to [Using Sysplex Distributor.](#)

## **Receiving on TCP using a queue sharing group**

Receiving shared channel programs are started in response to a startup request from the sending channel. To do so, a listener must be started to detect incoming network requests and start the associated channel. You start this listener program with the START LISTENER command, using the inbound disposition of the group, or using the operations and control panels.

All group listeners in the queue sharing group must be listening on the same port. If you have more than one channel initiator running on a single MVS image you can define virtual IP addresses and start your

TCP listener program to only listen on a specific address or host name by specifying IPADDR in the START LISTENER command. (For more information, see [START LISTENER](#).)

### **z/OS** Defining an LU 6.2 connection on z/OS

To define an LU 6.2 connection for a queue sharing group, certain attributes on the sending and receiving end must be configured.

For information about setting up APPC/MVS, see [Setting up communication for z/OS](#).

## **Connecting to APPC/MVS (LU 6.2)**

The connection name (CONNAME) field in the channel definition to connect to your queue sharing group must be set to the symbolic destination name, as specified in the side information data set for APPC/MVS. The partner LU specified in this symbolic destination must be the generic resource name. For more details, see [Defining yourself to the network using generic resources](#).

## **Receiving on LU 6.2 using a generic interface**

Receiving shared MCAs are started in response to a startup request from the sending channel. To do so, a group listener program must be started to detect incoming network requests and start the associated channel. The listener program is an APPC/MVS server. You start it with the START LISTENER command, using an inbound disposition group, or using the operations and control panels. You must specify the LU name to use a symbolic destination name defined in the side information data set. For more details, see [Defining yourself to the network using generic resources](#).

### **z/OS** Using IBM MQ with IMS

The IBM MQ -IMS adapter, and the IBM MQ - IMS bridge are the two components which allow IBM MQ to interact with IMS.

To configure IBM MQ and IMS to work together, you must complete the following tasks:

- [“Setting up the IMS adapter” on page 991](#)
- [“Setting up the IMS bridge” on page 998](#)

#### **Related concepts**

[IBM MQ and IMS](#)

[“Using IBM MQ with CICS” on page 999](#)

To use IBM MQ with CICS, you must configure the IBM MQ CICS adapter and, optionally, the IBM MQ CICS bridge components.

[“Using OTMA exits in IMS” on page 1001](#)

Use this topic if you want to use IMS Open Transaction Manager Access exits with IBM MQ for z/OS.

[IMS and IMS bridge applications on IBM MQ for z/OS](#)

#### **Related tasks**

[“Configuring queue managers on z/OS” on page 883](#)

Use these instructions to configure queue managers on IBM MQ for z/OS.

#### **Related reference**

[“Upgrading and applying service to Language Environment or z/OS Callable Services” on page 999](#)

The actions you must take vary according to whether you use CALLLIBS or LINK, and your version of SMP/E.

### **z/OS** Setting up the IMS adapter

To use IBM MQ within IMS requires the IBM MQ - IMS adapter (generally referred to as the IMS adapter).

This topic tells you how to make the IMS adapter available to your IMS subsystem. If you are not familiar with tailoring an IMS subsystem, see the [IMS documentation](#).

To make the IMS adapter available to IMS applications, follow these steps:

1. Define IBM MQ to IMS as an external subsystem using the IMS external subsystem attach facility (ESAF).

See [“Defining IBM MQ to IMS”](#) on page 993.

2. Include the IBM MQ load library thlqual.SCSQAUTH in the JOBLIB or STEPLIB concatenation in the JCL for your IMS control region and for any dependent region that connects to IBM MQ (if it is not in the LPA or link list). If your JOBLIB or STEPLIB is not authorized, also include it in the DFSESL concatenation after the library containing the IMS modules (usually IMS RESLIB).

Also include thlqual.SCSQANLx (where x is the language letter).

If DFSESL is present, then SCSQAUTH and SCSQANLx need to be included in the concatenation or added to LNKLIST. Adding to the STEPLIB or JOBLIB concatenation in the JCL is not sufficient.

3. Copy the IBM MQ assembler program CSQQDEFV from thlqual.SCSQASMS to a user library.
4. The supplied program, CSQQDEFV, contains one subsystem name CSQ1 identified as default with an IMS language interface token (LIT) of MQM1. You can retain this name for testing and installation verification.

For production subsystems, you change the NAME=CSQ1 to your own subsystem name, or use CSQ1. You can add further subsystem definitions as required. See [“Defining IBM MQ queue managers to the IMS adapter”](#) on page 996 for further information on LITs.

5. Assemble and link-edit the program to produce the CSQQDEFV load module. For the assembly, include the library thlqual.SCSQMACS in your SYSLIB concatenation; use the link-edit parameter RENT. This is shown in the sample JCL in thlqual.SCSQPROC(CSQ4DEFV).
6. Include the user library containing the module CSQQDEFV that you created in the JOBLIB or STEPLIB concatenation in the JCL for any dependent region that connects to IBM MQ. Put this library before the SCSQAUTH because SCSQAUTH has a default load module. If you do not do this, you will receive a user 3041 abend from IMS.
7. If the IMS adapter detects an unexpected IBM MQ error, it issues a z/OS SNAP dump to DD name CSQSNAP and issues reason code MQRC\_UNEXPECTED\_ERROR to the application. If the CSQSNAP DD statement was not in the IMS dependent region JCL, no dump is taken. If this happens, you could include the CSQSNAP DD statement in the JCL and rerun the application. However, because some problems might be intermittent, it is recommended that you include the CSQSNAP DD statement to capture the reason for failure at the time it occurs.
8. If you want to use dynamic IBM MQ calls (described in [Dynamically calling the IBM MQ stub](#)), build the dynamic stub, as shown in [Figure 114](#) on page 993.
9. If you want to use the IMS trigger monitor, define the IMS trigger monitor application CSQQTRMN, and perform PSBGEN and ACBGEN. See [“Setting up the IMS trigger monitor”](#) on page 997.
10. If you are using RACF to protect resources in the OPERCMDS class, ensure that the userid associated with your IBM MQ queue manager address space has authority to issue the MODIFY command to any IMS system to which it might connect.



```

//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD SYSOUT=*
//ACSQMOD DD DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB DD DISP=SHR,DSN=ims.reslib
//SYSLMOD DD DISP=SHR,DSN=private.load1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN DD *
INCLUDE ACSQMOD(CSQSTUB)
INCLUDE IMSLIB(DFSLI000)
ALIAS MQCONN,MQCONN,MQDISC MQI entry points
ALIAS MQGET,MQPUT,MQPUT1 MQI entry points
ALIAS MQOPEN,MQCLOSE MQI entry points
ALIAS MQBACK,MQCMIT MQI entry points
ALIAS CSQBBAK,CSQBCMT MQI entry points
ALIAS MQINQ,MQSET MQI entry points
ALIAS DFSPLI,PLITDLI IMS entry points
ALIAS DFSCOBOL,CBLTDLI IMS entry points
ALIAS DFSFOR,FORTDLI IMS entry points
ALIAS DFSASM,ASMTDLI IMS entry points
ALIAS DFSPASCL,PASTDLI IMS entry points
ALIAS DFHEI01,DFHEI1 IMS entry points
ALIAS DFSAIBLI,AIBTDLI IMS entry points
ALIAS DFSESS,DSNWLI,DSNHLI IMS entry points
ALIAS MQCRTMH,MQDLTMH,MQDLTMP IMS entry points
ALIAS MQINQMP,MQSETMP,MQMHBUF,MQBUFMH IMS entry points
MODE AMODE(31),RMODE(24) Note RMODE setting
NAME CSQDYNS(R)
/*

1Specify the name of a library accessible to IMS applications that
want to make dynamic calls to IBM MQ.

```

Figure 114. Sample JCL to link-edit the dynamic call stub

## Related concepts

IBM MQ and IMS

“Setting up the IMS bridge” on page 998

The IBM MQ - IMS bridge is an optional component that enables IBM MQ to input and output to and from existing programs and transactions that are not IBM MQ-enabled.

IMS and IMS bridge applications on IBM MQ for z/OS

## Defining IBM MQ to IMS

IBM MQ must be defined to the IMS control region, and to each dependent region accessing that IBM MQ queue manager. To do this, you must create a subsystem member (SSM) in the IMS.PROCLIB library, and identify the SSM to the applicable IMS regions.

## Placing the subsystem member entry in IMS.PROCLIB

Each SSM entry in IMS.PROCLIB defines a connection from an IMS region to a different queue manager.

To name an SSM, concatenate the value (one to four alphanumeric characters) of the IMSID field of the IMS IMCTRL macro with any name (one to four alphanumeric characters) defined by your site.

One SSM can be shared by all the IMS regions, or a specific member can be defined for each region.

This member contains as many entries as there are connections to external subsystems. Each entry is an 80-character record.

## Positional parameters

The fields in this entry are:

```
SSN,LIT,ESMT,RTT,REO,CRC
```

where:

**SSN**

Specifies the IBM MQ queue manager name. It is required, and must contain one through four characters.

**LIT**

Specifies the language interface token (LIT) supplied to IMS. This field is required, its value must match one in the CSQQDEFV module.

**ESMT**

Specifies the external subsystem module table (ESMT). This table specifies which attachment modules must be loaded by IMS. CSQQESMT is the required value for this field.

**RTT**

This option is not supported by IBM MQ.

**REO**

Specifies the region error option (REO) to be used if an IMS application references a non-operational external subsystem or if resources are unavailable at create thread time. This field is optional and contains a single character, which can be:

**R**

Passes a return code to the application, indicating that the request for IBM MQ services failed.

**Q**

Ends the application with an abend code U3051, backs out activity to the last commit point, does a PSTOP of the transaction, and requeues the input message. This option only applies when an IMS application tries to reference a non-operational external subsystem or if the resources are unavailable at create thread time.

IBM MQ completion and reason codes are returned to the application if the IBM MQ problem occurs while IBM MQ is processing the request; that is, after the adapter has passed the request on to IBM MQ.

**A**

Ends the application with an abend code of U3047 and discards the input message. This option only applies when an IMS application references a non-operational external subsystem or if the resources are unavailable at create thread time.

IBM MQ completion and reason codes are returned to the application if the IBM MQ problem occurs while IBM MQ is processing the request; that is, after the adapter has passed the request on to IBM MQ.

**CRC**

This option can be specified but is not used by IBM MQ.

**Note:** For full details of all positional parameters refer to [How external subsystems are specified to IMS](#).

An example SSM entry is:

```
CSQ1,MQM1,CSQQESMT,,R,
```

where:

**CSQ1**

The default subsystem name as supplied with IBM MQ. You can change this to suit your installation.

<b>MQM1</b>	The default LIT as supplied in CSQQDEFV.
<b>CSQQESMT</b>	The external subsystem module name. You must use this value.
<b>R</b>	REO option.

### Keyword parameters

IBM MQ parameters can be specified in keyword format. The SST parameter can have a value of either DB2 or MQ. Support for the MQ value was added in IMS 14. Use of MQ aids clarity, and the IMS subsystem command now includes the SST value, but otherwise does not have any significant effect. A value of DB2 can still be used if required. Other parameters are as described in [Positional parameters](#), and shown in the following example:

```
SST=MQ ,SSN=SYS3 ,LIT=MQM3 ,ESMT=CSQQESMT
```

where:

<b>SYS3</b>	The subsystem name
<b>MQM3</b>	The LIT as supplied in CSQQDEFV
<b>CSQQESMT</b>	The external subsystem module name

### Specifying the SSM EXEC parameter

Specify the SSM EXEC parameter in the startup procedure of the IMS control region. This parameter specifies the one-character to four-character subsystem member name (SSM).

If you specify the SSM for the IMS control region, any dependent region running under the control region can attach to the IBM MQ queue manager named in the IMS.PROCLIB member specified by the SSM parameter. The IMS.PROCLIB member name is the IMS ID (IMSID= *xxxx*) concatenated with the one to four characters specified in the SSM EXEC parameter. The IMS ID is the IMSID parameter of the IMSCTRL generation macro.

IMS lets you define as many external subsystem connections as are required. More than one connection can be defined for different IBM MQ queue managers. All IBM MQ connections must be within the same z/OS system. For a dependent region, you can specify a dependent region SSM or use the one specified for the control region. You can specify different region error options (REOs) in the dependent region SSM and the control region SSM. [Table 68 on page 995](#) shows the different possibilities of SSM specifications.

SSM for control region	SSM for dependent region	Action	Comments
No	No	None	No external subsystem can be connected.
No	Yes	None	No external subsystem can be connected.
Yes	No	Use the control region SSM	Applications scheduled in the region can access external subsystems identified in the control region SSM. Exits and control blocks for each attachment are loaded into the control region and the dependent region address spaces.
Yes	Yes (empty)	No SSM is used for the dependent region	Applications scheduled in this region can access DL/I databases only. Exits and control blocks for each attachment are loaded into the control region address space.

Table 68. SSM specifications options (continued)

SSM for control region	SSM for dependent region	Action	Comments
Yes	Yes (not empty)	Check the dependent region SSM with the control region SSM	Applications scheduled in this region can access only external subsystems identified in both SSMs. Exits and control blocks for each attachment are loaded into the control region and the dependent region address spaces.

There is no specific parameter to control the maximum number of SSM specification possibilities.

## Preloading the IMS adapter

The performance of the IMS adapter can be improved if it is preloaded by IMS. Preloading is controlled by the DFSMPLxx member of IMS.PROCLIB: see "IMS Administration Guide: System" for more information. The IBM MQ module names to specify are:

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQBACK	CSQQCMMT
CSQQESMT	CSQQPREP	CSQQTTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQFSTAB	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQTMID	CSQQTRGI	CSQQCON2	CSQBPAPI	CSQBCRMH	CSQBAPPL

For more information on the use of IBM MQ classes for JMS, see [Using IBM MQ classes for JMS in IMS](#).

Current releases of IMS support preloading IBM MQ modules from PDS-E format libraries in MPP, BMP, IFP, JMP and JBP regions only. Any other type of IMS region does not support preloading from PDS-E libraries. If preloading is required for any other type of region, then the IBM MQ modules that are provided must be copied to a PDS format library.

## Defining IBM MQ queue managers to the IMS adapter

The names of the IBM MQ queue managers and their corresponding language interface tokens (LITs) must be defined in the queue manager definition table.

Use the supplied CSQQDEFX macro to create the CSQQDEFV load module. [Figure 115 on page 996](#) shows the syntax of this assembler macro.

```
CSQQDEFX TYPE=ENTRY|DEFAULT,NAME=qmgr-name,LIT=token
or
CSQQDEFX TYPE=END
```

Figure 115. CSQQDEFX macro syntax

### Parameters

#### TYPE=ENTRY|DEFAULT

Specify either TYPE=ENTRY or TYPE=DEFAULT as follows:

#### TYPE=ENTRY

Specifies that a table entry describing an IBM MQ queue manager available to an IMS application is to be generated. If this is the first entry, the table header is also generated, including a CSQQDEFV CSECT statement.

**TYPE=DEFAULT**

As for TYPE=ENTRY. The queue manager specified is the default queue manager to be used when MQCONN or MQCONNX specifies a name that is all blanks. There must be only one such entry in the table.

**NAME= qmgr-name**

Specifies the name of the queue manager, as specified with **MQCONN** or **MQCONNX**.

**LIT= token**

Specifies the name of the language interface token (LIT) that IMS uses to identify the queue manager.

An MQCONN or MQCONNX call associates the *name* input parameter and the *hconn* output parameter with the name label and, therefore, the LIT in the CSQQDEFV entry. Further IBM MQ calls passing the *hconn* parameter use the LIT from the CSQQDEFV entry identified in the MQCONN or MQCONNX call to direct calls to the IBM MQ queue manager defined in the IMS SSM PROCLIB member with that same LIT.

In summary, the **name** parameter on the MQCONN or MQCONNX call identifies a LIT in CSQQDEFV and the same LIT in the SSM member identifies an IBM MQ queue manager. (For information about the MQCONN call, see [MQCONN - Connect queue manager](#). For information about the MQCONNX call, see [MQCONNX - Connect queue manager \(extended\)](#).)

**TYPE=END**

Specifies that the table is complete. If this parameter is omitted, TYPE=ENTRY is assumed.

**Using the CSQQDEFX macro**

Figure 116 on page 997 shows the general layout of a queue manager definition table.

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

Figure 116. Layout of a queue manager definition table

## **Setting up the IMS trigger monitor**

You can set up an IMS batch-oriented program to monitor an IBM MQ initiation queue.

Define the application to IMS using the model CSQQTAPL in the thlqual.SCSQPROC library (see [Example transaction definition for CSQQTRMN](#)).

Generate the PSB and ACB using the model CSQQTPSB in the thlqual.SCSQPROC library (see [Example PSB definition for CSQQTRMN](#)).

```
* This is the application definition *
* for the IMS Trigger Monitor BMP *

APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

Figure 117. Example transaction definition for CSQQTRMN

```

PCB TYPE=TP, ALTPCB for transaction messages
MODIFY=YES, To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB TYPE=TP, ALTPCB for diagnostic messages
MODIFY=YES, To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN, Runs program CSQQTRMN
CMPAT=YES

```

Figure 118. Example PSB definition for CSQQTRMN

For further information about starting and stopping the IMS trigger monitor, see [Controlling the IMS trigger monitor](#).

## Setting up the IMS bridge

The IBM MQ - IMS bridge is an optional component that enables IBM MQ to input and output to and from existing programs and transactions that are not IBM MQ-enabled.

This topic describes what you must do to customize the IBM MQ - IMS bridge.

### Define the XCF and OTMA parameters for IBM MQ.

This step defines the XCF group and member names for your IBM MQ system, and other OTMA parameters. IBM MQ and IMS must belong to the same XCF group. Use the OTMACON keyword of the CSQ6SYSP macro to tailor these parameters in the system parameter load module.

See [Using CSQ6SYSP](#) for more information.

### Define the XCF and OTMA parameters to IMS.

This step defines the XCF group and member names for the IMS system. IMS and IBM MQ must belong to the same XCF group.

Add the following parameters to your IMS parameter list, either in your JCL or in member DFSPBxxx in the IMS PROCLIB:

#### **OTMA=Y**

This starts OTMA automatically when IMS is started. (It is optional, if you specify OTMA=N you can also start OTMA by issuing the IMS command /START OTMA.)

#### **GRNAME=**

This parameter gives the XCF group name.

It is the same as the group name specified in the storage class definition (see the next step), and in the **Group** parameter of the OTMACON keyword of the CSQ6SYSP macro.

#### **OTMANM=**

This parameter gives the XCF member name of the IMS system.

This is the same as the member name specified in the storage class definition (see the next step).

### Tell IBM MQ the XCF group and member name of the IMS system.

This is specified by the storage class of a queue. If you want to send messages across the IBM MQ - IMS bridge you must specify this when you define the storage class for the queue. In the storage class, you must define the XCF group and the member name of the target IMS system. To do this, either use the IBM MQ operations and control panels, or use the IBM MQ commands as described in [Introduction to Programmable Command Formats](#).

### Set up the security that you require.

The /SECURE OTMA IMS command determines the level of security to be applied to **every** IBM MQ queue manager that connects to IMS through OTMA. See [Security considerations for using IBM MQ with IMS](#) for more information.

## Adding an additional IMS connection to the same queue manager

To add an IMS connection to the same queue manager you must define a second storage class (STGCLASS) to point at the new IMS; see [DEFINE STGCLASS](#) for more information.

### Important:

- One local queue cannot point to two storage classes.
- One storage class cannot point to two IMS bridges.
- IBM MQ and IMS must belong to the same XCF group. Use the OTMACON keyword of the CSQ6SYSP macro to tailor these parameters in the system parameter load module.

See [Using CSQ6SYSP](#) for more information.

### Related concepts

[IBM MQ and IMS](#)

[“Setting up the IMS adapter” on page 991](#)

To use IBM MQ within IMS requires the IBM MQ - IMS adapter (generally referred to as the IMS adapter).

[IMS and IMS bridge applications on IBM MQ for z/OS](#)

## Using IBM MQ with CICS

To use IBM MQ with CICS, you must configure the IBM MQ CICS adapter and, optionally, the IBM MQ CICS bridge components.

For more information about configuring the IBM MQ CICS adapter and the IBM MQ CICS bridge components, see the [Configuring connections to MQ](#) section of the CICS documentation.

### Related concepts

[IBM MQ and CICS](#)

[“Using IBM MQ with IMS” on page 991](#)

The IBM MQ -IMS adapter, and the IBM MQ - IMS bridge are the two components which allow IBM MQ to interact with IMS.

### Related reference

[“Upgrading and applying service to Language Environment or z/OS Callable Services” on page 999](#)

The actions you must take vary according to whether you use CALLLIBS or LINK, and your version of SMP/E.

## Upgrading and applying service to Language Environment or z/OS Callable Services

The actions you must take vary according to whether you use CALLLIBS or LINK, and your version of SMP/E.

The following tables show you what you need to do to IBM MQ for z/OS if you upgrade your level of, or apply service to, the following products:

- Language Environment
- z/OS Callable Services (APPC and RRS for example)

Table 69. Service has been applied or the product has been upgraded to a new release

Product	Action if using CALLLIBS and SMP/E V3r2 or later  <b>Note: You do not need to run separate jobs for Language Environment and Callable services. One job will suffice.</b>	Action if using LINK
Language Environment	<ol style="list-style-type: none"> <li>1. Set the Boundary on your SMP/E job to the Target zone.</li> <li>2. On the SMP_CNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>z/OS SMP/E Commands</i> for further information.</li> <li>3. Run the SMP/E job.</li> </ol>	No action required provided that the SMP/E zones were set up for automatic relinking, and the CSQ8SLDQ job has been run.
Callable Services	<ol style="list-style-type: none"> <li>1. Set the Boundary on your SMP/E job to the Target zone.</li> <li>2. On the SMP_CNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>z/OS SMP/E Commands</i> for further information.</li> <li>3. Run the SMP/E job.</li> </ol>	No action required provided that the SMP/E zones were set up for automatic relinking, and the CSQ8SLDQ job has been run.

Table 70. One of the products has been updated to a new release in a new SMP/E environment and libraries

Product	Action if using CALLLIBS and SMP/E V3r2 or later  <b>Note: You do not need to run three separate jobs for Language Environment and Callable services. One job will suffice for both products.</b>	Action if using LINK
Language Environment	<ol style="list-style-type: none"> <li>1. Change the DDDEFs for SCEELKED and SCEESPC to point to the new library.</li> <li>2. Set the Boundary on your SMP/E job to the Target zone.</li> <li>3. On the SMP_CNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>z/OS SMP/E Commands</i> for further information.</li> <li>4. Run the SMP/E job.</li> </ol>	<ol style="list-style-type: none"> <li>1. Delete the XZMOD subentries for the following LMOD entries in the IBM MQ for z/OS target zone: CMQXDCST, CMQXRCTL, CMQXSUPR, CSQCBE00, CSQCBE30, CSQCBP00, CSQCBP10, CSQCBR00, CSQUCVX, CSQUDLQH, CSQVXPCB, CSQVXSPT, CSQXDCST, CSQXRCTL, CSQXSUPR, CSQXTDMI, CSQXTCP, CSQXTNSV, CSQ7DRPS, IMQB23IC, IMQB23IM, IMQB23IR, IMQS23IC, IMQS23IM, IMQS23IR</li> <li>2. Set up the appropriate ZONEINDEXs between the IBM MQ zones and the Language Environment zones.</li> <li>3. Tailor CSQ8SLDQ to refer to the new zone on the FROMZONE parameter of the LINK commands. CSQ8SLDQ can be found in the SCSQINST library.</li> <li>4. Run CSQ8SLDQ.</li> </ol>



Table 70. One of the products has been updated to a new release in a new SMP/E environment and libraries (continued)

Product	Action if using CALLLIBS and SMP/E V3r2 or later  <b>Note: You do not need to run three separate jobs for Language Environment and Callable services. One job will suffice for both products.</b>	Action if using LINK
Callable services	<ol style="list-style-type: none"> <li>1. Change the DDDEF for CSSLIB to point to the new library</li> <li>2. Set the Boundary on your SMP/E job to the Target zone.</li> <li>3. On the SMP_CNTL card specify LINK LMODS CALLLIBS. You can also specify other parameters such as CHECK, RETRY(YES) and RC. See <i>z/OS SMP/E Commands</i> for further information.</li> <li>4. Run the SMP/E job.</li> </ol>	<ol style="list-style-type: none"> <li>1. Delete the XZMOD subentries for the following LMOD entries in the IBM MQ for z/OS target zone:  CMQXRCTL, CMQXSUPR, CSQBSRV, CSQILPLM, CSQXJST, CSQXRCTL, CSQXSUPR, CSQ3AMGP, CSQ3EPX, CSQ3REPL</li> <li>2. Set up the appropriate ZONEINDEXs between the IBM MQ zones and the Callable Services zones.</li> <li>3. Tailor CSQ8SLDQ to refer to the new zone on the FROMZONE parameter of the LINK commands. CSQ8SLDQ can be found in the SCSQINST library.</li> <li>4. Run CSQ8SLDQ.</li> </ol>

For an example of a job to relink modules when using CALLLIBS, see [“Running a LINK CALLLIBS job”](#) on page 1001.

### Running a LINK CALLLIBS job

An example job to relink modules when using CALLLIBS.

The following is an example of the job to relink modules when using CALLLIBs on a SMP/E V3r2 system. You must provide a JOBCARD and the data set name of SMP/E CSI that contains IBM MQ for z/OS.

```

//*****
//* RUN LINK CALLLIBS.
//*****
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI DD DSN=your.csi
// DISP=SHR
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZONE).
LINK LMODS CALLLIBS .
/*

```

Figure 119. Example SMP/E LINK CALLLIBS job

### Using OTMA exits in IMS

Use this topic if you want to use IMS Open Transaction Manager Access exits with IBM MQ for z/OS.

If you want to send output from an IMS transaction to IBM MQ, and that transaction did not originate in IBM MQ, you need to code one or more IMS OTMA exits.

Similarly if you want to send output to a non-OTMA destination, and the transaction did originate in IBM MQ, you also need to code one or more IMS OTMA exits.

The following exits are available in IMS to enable you to customize processing between IMS and IBM MQ:

- An OTMA pre-routing exit
- A destination resolution user (DRU) exit

## OTMA exit names

You must name the pre-routing exit DFSYPRX0. You can name the DRU exit anything, as long as it does not conflict with a module name already in IMS.

### Specifying the destination resolution user exit name

You can use the *Druexit* parameter of the OTMACON keyword of the CSQ6SYSP macro to specify the name of the OTMA DRU exit to be run by IMS.

To simplify object identification, consider adopting a naming convention of DRU0xxxx, where xxxx is the name of your IBM MQ queue manager.

If you do not specify the name of a DRU exit in the OTMACON parameter, the default is DFSYDRU0. See [DFSYDRU0](#) for more information.

### Naming convention for IMS destination

You need a naming convention for the destination to which you send the output from your IMS program. This is the destination that is set in the CHNG call of your IMS application, or that is preset in the IMS PSB.

## A sample scenario for an OTMA exit

Use the following topics for an example of a pre-routing exit and a destination routing exit for IMS:

- [“The pre-routing exit DFSYPRX0” on page 1002](#)
- [“The destination resolution user exit” on page 1003](#)

To simplify identification, make the OTMA destination name similar to the IBM MQ queue manager name, for example the IBM MQ queue manager name repeated. In this case, if the IBM MQ queue manager name is " **VCPE** ", the destination set by the CHNG call is " **VCPEVCPE** ".

### Related concepts

[IBM MQ and IMS](#)

[“Using IBM MQ with IMS” on page 991](#)

The IBM MQ -IMS adapter, and the IBM MQ - IMS bridge are the two components which allow IBM MQ to interact with IMS.

[IMS and IMS bridge applications on IBM MQ for z/OS](#)

## The pre-routing exit DFSYPRX0

This topic contains a sample pre-routing exit for OTMA in IMS.

You must first code a pre-routing exit DFSYPRX0. See [OTMA Destination Resolution user exit \(DFSYPRX0 and other OTMAYPRX type exits\)](#) for parameters passed to this routine by IMS.

This exit tests whether the message is intended for a known OTMA destination (in our example VCPEVCPE). If it is, the exit must check whether the transaction sending the message originated in OTMA. If the message originated in OTMA, it will have an OTMA header, so you should exit from DFSYPRX0 with register 15 set to zero.

- If the transaction sending the message did not originate in OTMA, you must set the client name to be a valid OTMA client. This is the XCF member-name of the IBM MQ queue manager to which you want to send the message. You should set your client name (in the OTMACON parameter of the CSQ6SYSP

macro) is set to the queue manager name. This is the default. You should then exit from DFSYPRX0 setting register 15 to 4.

- If the transaction sending the message originated in OTMA, and the destination is non-OTMA, you should set register 15 to 8 and exit.
- In all other cases, you should set register 15 to zero.

If you set the OTMA client name to one that is not known to IMS, your application CHNG or ISRT call returns an A1 status code.

For an IMS system communicating with more than one IBM MQ queue manager, you should repeat the logic for each IBM MQ queue manager.

Sample assembler code is shown in [Figure 120 on page 1003](#):

```
TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTEMTIME
SPACE 2
LR R12,R15 MODULE ADDRESSABILITY
USING DFSYPRX0,R12
*
L R2,12(,R1) R2 -> OTMA PREROUTE PARMS
*
LA R3,48(,R2) R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC 0(16,R3),=XL16'00' OTMA ORIG?
BNE OTMAIN YES, GO TO THAT CODE
*
NOOTMAIN DS 0H NOT OTMA INPUT
LA R5,8(,R2) R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT0 NO, NORMAL PROCESSING
*
L R4,80(,R2) R4 AT ADDR OF OTMA CLIENT
MVC 0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B EXIT4 AND EXIT
*
OTMAIN DS 0H OTMA INPUT
LA R5,8(,R2) R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT8 NO, NORMAL PROCESSING

*
EXIT0 DS 0H
LA R15,0 RC = 0
B BYEBYE
*
EXIT4 DS 0H
LA R15,4 RC = 4
B BYEBYE
*
EXIT8 DS 0H
LA R15,8 RC = 8
B BYEBYE
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END
```

*Figure 120. OTMA pre-routing exit assembler sample*

## The destination resolution user exit

This topic contains a sample destination resolution user exit for IMS.

If you have set registers 15 to 4 in DFSYPRX0, or if the source of the transaction was OTMA **and** you set Register 15 to zero, your DRU exit is invoked. In this example, the DRU exit name is DRU0VCPE.

The DRU exit checks if the destination is VCPEVCPE. If it is, it sets the OTMA user data (in the OTMA prefix) as follows:

**Offset**

**OTMA user data**

**(decimal)**

**0**

OTMA user data length (in this example, 334)

**2**

MQMD

**326**

Reply to format

These offsets are where the IBM MQ - IMS bridge expects to find this information.

The DRU exit should be as simple as possible. Therefore, in this sample, all messages originating in IMS for a particular IBM MQ queue manager are put to the same IBM MQ queue.

If the message needs to be persistent, IMS must use a synchronized transaction pipe. To do this, the DRU exit must set the OUTPUT flag. See [Specifying synchronized tpipes for IBM MQ](#) for more information.

Write an IBM MQ application to process this queue, and use information from the MQMD structure, the MQIIH structure (if present), or the user data, to route each message to its destination.

A sample assembler DRU exit is shown in [Figure 121 on page 1005](#).

```

TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTEME
SPACE 2
LR R12,R15 MODULE ADDRESSABILITY
USING DRU0VCPE,R12
*
L R2,12(,R1) R2 -> OTMA DRU PARMS
*
L R5,88(,R2) R5 ADDR OF OTMA USERDATA
LA R6,2(,R5) R6 ADDR OF MQMD
USING MQMD,R6 AS A BASE
*
LA R4,MQMD_LENGTH+10 SET THE OTMA USERDATA LEN
STH R4,0(,R5) = LL + MQMD + 8
*
 CLEAR REST OF USERDATA
MVI 0(R6),X'00' ...NULL FIRST BYTE
MVC 1(255,R6),0(R6) ...AND PROPAGATE IT
MVC 256(MQMD_LENGTH-256+8,R6),255(R6) ...AND PROPAGATE IT
*
VCPE DS 0H
CLC 44(16,R2),=CL16'VCPE' IS DESTINATION VCPE?
BNE EXIT4 NO, THEN DEST IS NON-OTMA
MVC MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC MQMD_REPLYTOQMGR,=CL48'VCPE' SET QNAME AND QMGRNAME
MVC MQMD_FORMAT,MQFMT_IMS SET MQMD FORMAT NAME
MVC MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*
 SET REPLYTO FORMAT NAME
B EXIT0
*
EXIT0 DS 0H
LA R15,0 SET RC TO OTMA PROCESS
B BYEBYE AND EXIT
*
EXIT4 DS 0H
LA R15,4 SET RC TO NON-OTMA
B BYEBYE AND EXIT
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END

```

Figure 121. Sample assembler DRU exit

z/OS

## Using IBM z/OSMF to automate IBM MQ

The IBM z/OS Management Facility (z/OSMF) provides system management functions in a task-oriented, web browser-based user interface with integrated user assistance, so that you can more easily manage the day-to-day operations and administration of your mainframe z/OS systems.

By streamlining some traditional tasks and automating others, z/OSMF can help to simplify some areas of z/OS system management.

Resources can be provisioned or de-provisioned, at a click of a button, from a user provided portal. z/OSMF provides REST APIs to help with this task.

The sample marketplace portal supplied with z/OSMF can also be used to provision and de-provision resources. Alternatively, more experienced users can use the z/OSMF Web User Interface (WUI).

This section assumes that you understand z/OSMF, but if you are unfamiliar with z/OSMF you should read [Getting started with z/OSMF](#). Alternatively, you can access this section from the z/OSMF WUI online help.

You should familiarize yourself with z/OS Cloud configuration, that is:

- Cloud Provisioning - [Resource management services](#)
- Workload Management - see [IBM z/OS Management Facility Programming Guide](#) for more information.
- Getting started - see [Getting Started Tutorial - Cloud](#)

z/OSMF 2.2 introduces role based activities and tasks, so it is important that you understand concepts like:

- domains
- administrators
- approvers
- tenants
- templates
- instances
- workflows

and so on.

Sample IBM MQ z/OSMF workflows and associated files are provided, and can be installed as part of the IBM MQ for z/OS UNIX System Services Components feature. The installation process for this feature, and the directory and file structure, are described in the IBM MQ for z/OS Program Directory. Per i collegamenti di download per le directory del programma, consultare [IBM MQ for z/OS Program Directory PDF files](#).

The sample workflows are written in XML and demonstrate how to automate the provisioning (creation) or de-provisioning (destruction) of IBM MQ queue managers, channel initiators, and local queues, and how to perform actions against the provisioned IBM MQ resources. Steps within the workflows submit jobs (JCL), run REXX execs, process Shell scripts, or issue REST API calls.

The samples are designed to illustrate the types of function that can be achieved using z/OSMF. It is anticipated that z/OSMF workflows will generally be used to provision resources and actions like put or get message will, in essence, be performed using IBM MQ applications.

You can run the sample workflows as supplied, provided the workflow variable properties have been set (as discussed in the following sections), or you can customize them as required. You might prefer to write your own workflows to perform additional function. Before running the sample workflows see:

- [“Prerequisites for z/OSMF” on page 1006](#)
- [“Security settings ” on page 1008](#)
- [“Limitations ” on page 1010](#)

Sample workflow applications are provided to:

- [“Automate the provisioning or de-provisioning of IBM MQ queue managers and perform actions against the provisioned queue managers” on page 1011](#)
- [“Automate the provisioning or de-provisioning of IBM MQ local queues and perform actions against the provisioned queues” on page 1012.](#)

### **Related concepts**

[“Setting up IBM MQ for z/OS” on page 888](#)

Use this topic as a step by step guide for customizing your IBM MQ for z/OS system .

## **Prerequisites for z/OSMF**

The prerequisites you require to run IBM z/OS Management Facility (z/OSMF) with IBM MQ

The workflows shipped in IBM MQ for z/OS 9.1.0 exploit new function in z/OSMF, which is provided through APARs on both z/OS 2.1 and 2.2. More details are provided in the following text.

1. You have installed and configured IBM z/OS Management Facility 2.2 correctly. If you are running with security enabled, ensure that all security settings as documented by z/OSMF have been configured.

2. You have installed the following APARs for:

**z/OS 2.1**

- PI71068
- PI71079
- PI71082
- PI71084
- OA50130

**z/OS 2.2**

- PI70526
- PI70521
- PI70527
- PI67839
- PI70767
- PI46315
- OA49081
- OA49802
- OA50130

3. The z/OSMF angel (if required) and server processes have been configured.

4. The z/OS Cloud environment has been configured (as briefly discussed above and documented by z/OSMF)

5. IBM MQ for z/OS 9.0.1 has been installed and the product load libraries are available.

6. The following IBM MQ queue manager customization tasks have been performed:

Task	Description
1	Identify the z/OS system parameters
2	APF authorize the IBM MQ load libraries
3	Update the z/OS link list and LPA
4	Update the z/OS program properties table

7. The sample workflows and associated files are installed in a suitable z/OS UNIX System Services (z/OS UNIX) directory.

8. The /tmp z/OS UNIX directory is available, because the provision.xml workflow might create a temporary file in this directory. If a file is created, the workflow, in general, deletes the file after use.

9. The deprovision.xml file has steps in it that invoke the CSQ4ZWS1.rexx and CSQ4ZWS2.rexx REXX execs. These execs wait for the queue manager and channel initiator subsystems to stop; the execs invoke the z/OS UNIX **SLEEP** command as a system call.

Depending on your z/OS UNIX configuration, you might find that the **SLEEP** command does not work as coded. If, during processing you encounter an error which indicates that the **SLEEP** command cannot be found, you can try replacing the following lines in execs CSQ4ZWS1.rexx and CSQ4ZWS2.rexx:

```
CALL SYSCALLS('ON') /* Enable z/OS UNIX calls */
ADDRESS SYSCALL
```

```
"SLEEP" 10 /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF' /* Disable z/OS UNIX calls */
```

with

```
'sleep' 10
```

Then, issue the Open MVS (OMVS) **env** command to check your PATH environment variable setting. Ensure that the directory which contains the **sleep** command is defined to the PATH. Note that the **sleep** command is typically found in the /bin directory.

10. Ensure that z/OSMF has been started.

Both the angel and server z/OSMF processes must be started and the z/OSMF Web User Interface (WUI) be up and running. For further details, see [Liberty profile: Process types on z/OS](#).

Even if you intend to drive the workflows using the REST API, the z/OSMF WUI needs to be started. The z/OSMF WUI can be useful for monitoring the creation and execution of workflows.

### Related concepts


[“Using IBM z/OSMF to automate IBM MQ” on page 1005](#)

The IBM z/OS Management Facility (z/OSMF) provides system management functions in a task-oriented, web browser-based user interface with integrated user assistance, so that you can more easily manage the day-to-day operations and administration of your mainframe z/OS systems.

## Security settings

The security settings required to run z/OSMF.

The following User ID variable properties are defined in the properties file. For more details, see [“Running the workflows” on page 1015](#).

User ID property	Description
CSQ_USERID	User ID used to run the workflow steps. Note, however, that selected steps (which generally require an elevated level of authority) will be run with different user IDs based on the setting of the <b>CSQ_ADMIN_*</b> user IDs listed in the following text. The user ID in use is identified by the <b>runAsUser</b> property on the respective step in the workflows.
CSQ_ADMIN_APF_USERID	User ID to use when APF authorizing the load library that contains the queue manager system parameter module.
CSQ_APF_APPROVAL_ID	The approval ID used to permit users to run the data set APF authorization step as user CSQ_ADMIN_APF_USERID.
CSQ_ADMIN_CONSOLE_USERID	User ID used when running steps under the run that issue z/OS console commands.   <b>Attention:</b> This user ID needs to be permitted UPDATE access to the started task profile (MVS.START.STC.*) in the OPERCMDS class. See <a href="#">Controlling the use of operator commands in the z/OS documentation</a> for more information.
CSQ_CONSOLE_APPROVAL_ID	The approval ID used to permit users to run steps that issue z/OS console commands under the run as user CSQ_ADMIN_CONSOLE_USERID.
CSQ_ADMIN_SAF_USERID	User ID to use when issuing SAF commands.
CSQ_SAF_APPROVAL_ID	The approval ID used to permit users to run the SAF command steps under the run as user CSQ_ADMIN_SAF_USERID.
CSQ_ADMIN_SSI_USERID	User ID to use when issuing the SETSSI command to identify the subsystem being provisioned to z/OS.



User ID property	Description
CSQ_SSI_APPROVAL_ID	The approval ID used to permit users to run the SETSSI command step under the run as user CSQ_ADMIN_SSI_USERID.

**Note:** The User ID being used to run the provision and de-provision workflows needs to have sufficient authority as listed below:

1. The Queue Manager provision and de-provision workflows use the SETPROG command to APF authorize data sets. Either the user ID is set in property CSQ\_ADMIN\_APF\_USERID, or the user ID being used to run the workflows needs to be permitted to issue this command. You can achieve this by issuing the following command:

```
PERMIT MVS.SETPROG CLASS(OPERCMD5) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

**Note:** The SETPROG command might not persist across an IPL of a z/OS system so, it might be necessary to manually issue the following SETPROG command following an IPL:

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

For more details about the SETPROG command, see [Using RACF to control APF lists](#).

In addition, you might have enabled FACILITY class to control which libraries can be APF authorized, so you might need to issue the command:

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID)
ACCESS(UPDATE)
```

2. A step in the Queue Manager provision workflow issues the SETSSI command to identify the IBM MQ subsystem to z/OS. The User ID set in property CSQ\_ADMIN\_SSI\_USERID needs to be permitted to use this command. You can achieve this by issuing the following command:

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMD5) ID(value of CSQ_ADMIN_SSI_USERID)
ACCESS(CONTROL)
```

**Note:** Subsystems that have been identified to z/OS through the SETSSI command do not persist across an IPL of a z/OS system. So, it might be necessary to manually issue the following SETSSI command following an IPL:

```
SETSSI ADD,S=value of CSQ_SSID,I=CSQ3INI,
P=CSQ3EPX,value of CSQ_CMD_PFX,S'
```

For more details about the SETSSI command, see: [SETSSI command](#).

3. The workflows issue queue manager commands, so if you are planning to enable security, the user ID set in property CSQ\_ADMIN\_RACF\_USERID (or the user ID being used to run the workflows) needs to be granted CLAUTH (client authentication) authority to the MQADMIN or the MXADMIN class (depending on which class is being used). This is to allow this user ID to define security profiles to these classes. You can achieve this by issuing the following command:

```
ALTUSR value of CSQ_ADMIN_RACF_USERID CLAUTH(MQADMIN)
```

For more details about **CLAUTH** see [The CLAUTH \(class authority\) attribute](#).

4. The deprovision.xml workflow issues z/OS commands, for example, DISPLAY ACTIVE jobs, CANCEL or FORCE subsystems, so the user ID set in property CSQ\_ADMIN\_CONSOLE\_USERID (or the user ID being used to run the workflows) needs to have suitable authority to issue such commands.
5. Users requesting a queue manager instance, using the templates table of the Software Services task, must have permission to access z/OSMF and the Configuration Assistant, as defined by z/OSMF.

6. The user ID of the consumer provisioning a queue manager requires authority to add and delete members from the PROCLIB data set defined with variable CSQ\_PROC\_LIB.
7. A queue manager must be provisioned ahead of provisioning queues.
8. To use the queueLoad.xml and queueOffload.xml workflows, the data sets used need to be defined ahead of time. Also, the user ID used to run these workflows needs to be granted UPDATE authority to the data sets.
9. A step in the queue manager provision.xml workflow currently disables subsystem security. You can modify Job csq4znse.jcl to enable subsystem security by adding the appropriate security commands for protecting IBM MQ resources. However, note that if you do add additional commands, you also need to add commands to delete security permissions in csq4dse.jcl, which is submitted by the deprovision.xml workflow.

**Note:** This step issues RACF security commands. If you are using an alternate security product, you need to modify this step to issue the appropriate commands for your security product.

## Network Requirements

When adding a queue manager template, and resources for the template, you need to click **Create network resource pool**. This creates a resource pool with network resources for this template.

Using the Configuration Assistant, your network administrator needs to complete this network resource pool definition by defining a limit for the number of ports that are to be allocated for this template.

For each template instance, the provision.xml workflow allocates a port in the range, and starts a listener to listen on that port.

## Classifying with IBM Workload Manager

If you want to classify the queue manager and channel initiator address spaces with WLM, you need to specify this when adding a template for provisioning a queue manager.

Whether to classify or not, is controlled by flags **CSQ\_DEFINE\_MSTR\_WLM\_RULE** and **CSQ\_DEFINE\_CHIN\_WLM\_RULE**, which are set in file workflow\_variables.properties.

For more information about classifying with WLM, refer to the *z/OSMF Configuration Guide*.

### Related concepts

“Prerequisites for z/OSMF” on page 1006

The prerequisites you require to run IBM z/OS Management Facility (z/OSMF) with IBM MQ

## Limitations

Limitations when using z/OSMF with IBM MQ.

1. The provision.xml workflow currently automates the following highlighted queue manager customization tasks:

Task	Description
1	Identify the z/OS system parameters
2	APF authorize the IBM MQ load libraries ( <b>provision.xml does APF authorize some libraries</b> )
3	Update the z/OS link list and LPA
4	Update the z/OS program properties table
5	<b>Define the IBM MQ subsystem to z/OS</b>
6	<b>Create procedures for the IBM MQ queue manager</b>
7	<b>Create procedures for the channel initiator</b>

Task	Description
8	<b>Define the IBM MQ subsystem to a z/OS WLM service class</b>
9	Select and set up your coupling facility offload storage environment
10	Set up the coupling facility
11	Implement your ESM security controls
12	Update SYS1.PARMLIB members
13	<b>Customize the initialization input data sets</b>
14	<b>Create the bootstrap and log data sets</b>
15	<b>Define your page sets</b>
16	Add the IBM MQ entries to the Db2 data-sharing group
17	<b>Tailor your system parameter modules (some)</b>
18	<b>Tailor the channel initiator parameters (some)</b>
19	Set up Batch, TSO, and RRS adapters
20	Set up the operations and control panels
21	Include the IBM MQ dump formatting member
22	Suppress information messages
23	Update your system DIAG member for Advanced Message Security
24	Create procedures for Advanced Message Security
25	Set up the started task user Advanced Message Security
26	Grant RACDCERT permissions to the security administrator for Advanced Message Security
27	Grant users resource permissions for Advanced Message Security

2. Customization tasks that are not highlighted in bold text need to be performed manually, if required.
3. The sample INP1 and INP2 members are currently used as is. If required, additional properties can be defined to control the resources defined by these members.
4. Comments pertaining to specific properties listed in the properties file indicate any limitations of using those properties. For more details, see [“Running the workflows” on page 1015](#).

#### Related concepts

[“Security settings” on page 1008](#)

The security settings required to run z/OSMF.

### Automate the provisioning of IBM MQ objects

Samples are supplied to automate the provisioning of queue managers and local queues.

#### Automate the provisioning or de-provisioning of IBM MQ queue managers and perform actions against the provisioned queue managers

The following queue manager specific sample z/OSMF workflows are provided:

Workflow name	Description
provision.xml	Provision an IBM MQ for z/OS queue manager

Workflow name	Description
	<p>This sample workflow:</p> <ul style="list-style-type: none"> <li>• Provisions the required system resources for a queue manager.</li> <li>• Provisions the required system resources for a channel initiator.</li> <li>• Starts the queue manager (which also starts the channel initiator and TCP/IP listener)</li> <li>• Runs the sample queue manager installation verification program.</li> </ul> <p>An environment property can be set to control the provisioning of queue managers with different characteristics. For more information, see <a href="#">“Running the workflows” on page 1015</a>.</p> <p><b>Note:</b> A manifest file (<code>provision.mf</code>) is provided to assist with adding a template for this workflow. This file contains a reference to the <b>qaas_readme.pdf</b> file which contains additional information. You can access the file through a link, once the template has been added.</p>
deprovision.xml	<p>De-provision an IBM MQ for z/OS queue manager</p> <p>This sample workflow:</p> <ul style="list-style-type: none"> <li>• Stops the channel initiator (which also stops the TCP/IP listener) and the queue manager.</li> <li>• Waits for the subsystems to stop</li> <li>• De-provisions all channel initiator and queue manager system resources.</li> </ul>
startQMgr.xml	<p>Start an IBM MQ for z/OS queue manager</p> <p>This sample workflow starts the queue manager (which also starts the channel initiator and TCP/IP listener).</p>
stopQMgr.xml	<p>Stop an IBM MQ for z/OS queue manager</p> <p>This sample workflow stops the channel initiator (which also stops the TCP/IP listener) and the queue manager.</p>

Each workflow performs one or more steps. Comments in the workflows explain the function performed by each step. Some of the steps just request data input, while some steps submit JCL, invoke REXX execs, Shell scripts, or issue REST API calls to accomplish the stated function.

Refer to each step for the exact name of the JCL or REXX exec files. The workflows and associated JCL or REXX exec files reference variables that are declared in one or more variable XML files. For more details, see [“Workflow variable declaration files” on page 1014](#).

**deprovision**, **startQMgr**, and **stopQMgr** can be performed as actions against a provisioned IBM MQ for z/OS queue manager.

## Automate the provisioning or de-provisioning of IBM MQ local queues and perform actions against the provisioned queues

The following queue specific sample z/OSMF workflows are provided:

Workflow name	Description
defineQueue.xml	<p>Define a local queue</p> <p>This sample workflow demonstrates how z/OSMF workflows can be used to define small, medium, or large sized queues based on property settings.</p>

Workflow name	Description
	<p><b>Note:</b> A manifest file (provision.mf) is provided to assist with adding a template for this workflow. This file contains a reference to the <b>qaas_readme.pdf</b> file which contains additional information. You can access the file through a link, once the template has been added.</p>
displayQueue.xml	<p>Display selected attributes of a local queue</p> <p>This sample workflow displays selected attributes of a local queue. The attributes are returned in a z/OSMF variable (refer to the steps in the workflow for the name of the variable) and subsequently displayed. If required, the contents of the variable can be accessed using a REST API.</p> <p>For more details, refer to <a href="#">Cloud provisioning REST APIs</a>, and also see <a href="#">z/OSMF workflow services</a>.</p>
deleteQueue.xml	<p>Delete a local queue</p> <p>This sample workflow deletes a local queue on a specified queue manager.</p>
putQueue.xml	<p>Put one or more messages to a local queue.</p> <p>This sample workflow puts one or more messages to a local queue. The message text can be specified but if more than one message is put to a local queue at the same time, the same message text is used.</p>
getQueue.xml	<p>Get one or more messages from a local queue.</p> <p>This sample workflow gets one or more messages from a local queue. The messages are returned in a z/OSMF variable (refer to the steps in the workflow for the name of the variable) and subsequently displayed. If required, you can access the contents of the variable using a REST API.</p> <p>For more details, refer to <a href="#">Cloud provisioning REST APIs</a>, and also see <a href="#">z/OSMF workflow services</a>.</p>
loadQueue.xml	<p>Load messages from a data set to a local queue.</p> <p>This sample workflow loads messages from a data set on to a local queue. The default name of the data set is specified by setting a property. For more details, see <a href="#">“Running the workflows” on page 1015</a>.</p>
offloadQueue.xml	<p>Offload messages from a local queue to a data set.</p> <p>This sample workflow off-loads messages from a local queue to a data set. The default name of the data set is specified by setting a property. For more details, see <a href="#">“Running the workflows” on page 1015</a>.</p>
clearQueue.xml	<p>Clear messages on a local queue.</p> <p>This sample workflow clears (deletes) all messages on a local queue.</p>

**Notes:**

1. The **Put Queue** action allows you to enter some message data and put one or more messages onto a queue. If more than one message is to be placed onto a queue during a given request, the same message data is used.
2. The loadQueue.xml and offloadQueue.xml workflows invoke the executable module, CSQUDMSG in the SCSQLOAD library, with an alias of QLOAD. This is equivalent to the **dmpmqmsg** utility available with IBM MQ for Multiplatforms. Therefore messages loaded from a data set onto a queue, or from a queue onto a data set, are expected to be in the **dmpmqmsg** format.

Sample JCL is also provided as member CSQ4QL0D in SCSQPROC.

The easiest way to try out the loadQueue and offloadQueue actions is to do the following:

- a. Issue **putQueue** a few times to put some messages on to a queue.
- b. Use **offloadQueue** to offload the messages from the queue on to a data set.
- c. If required, issue **clearQueue** to remove all messages from the queue.
- d. Use **loadQueue** to load the messages from a data set onto the same or a different queue.

If you are interested in the **dmpmqmsg** format, you can browse the contents of the data set, once you have issued an Offload request.

3. You can perform **displayQueue**, **deleteQueue**, **putQueue**, **getQueue**, **loadQueue**, **offloadQueue**, and **clearQueue** as actions against a provisioned IBM MQ for z/OS local queue. For further details about actions and action files, refer to the *z/OSMF Programming Guide*.
4. All action related workflows are deleted by default. The reason for this is to minimize the need for users to cleanup workflows.

The problem with this however is that where an action results in some output. For example, the **displayQueue** and **getQueue** actions both produce output.

The output cannot be seen since the related workflow is deleted as soon as the action has been performed. So, if you drive the workflow actions from the z/OS WUI, you need to set the **cleanAfterComplete** flag to *false* on the **<workflow>** tag for each action whose output you want to see.

For example, to see the output of **displayQueue**, set the flag as follows:

```
<action name="displayQueue">
 <workflow cleanAfterComplete="false">
 ...
 </workflow>
</action>
```

However, this means that you then have to manually clean up action related workflows.

Each sample z/OSMF workflow performs one or more steps. Comments in the workflows explain the function performed by each step. Some of the steps just request data input while some steps submit JCL and others invoke REXX execs to accomplish the stated function.

Refer to each step for the exact name of the JCL or REXX exec files. The workflows and associated JCL or REXX exec files reference variables that are declared in one or more [“Workflow variable declaration files”](#) on page 1014.

### Related concepts

[“Limitations ” on page 1010](#)

Limitations when using z/OSMF with IBM MQ.

## Running workflows

A description of the files referenced by the sample The z/OSMF workflows, and how you run a workflow.

### Workflow variable declaration files

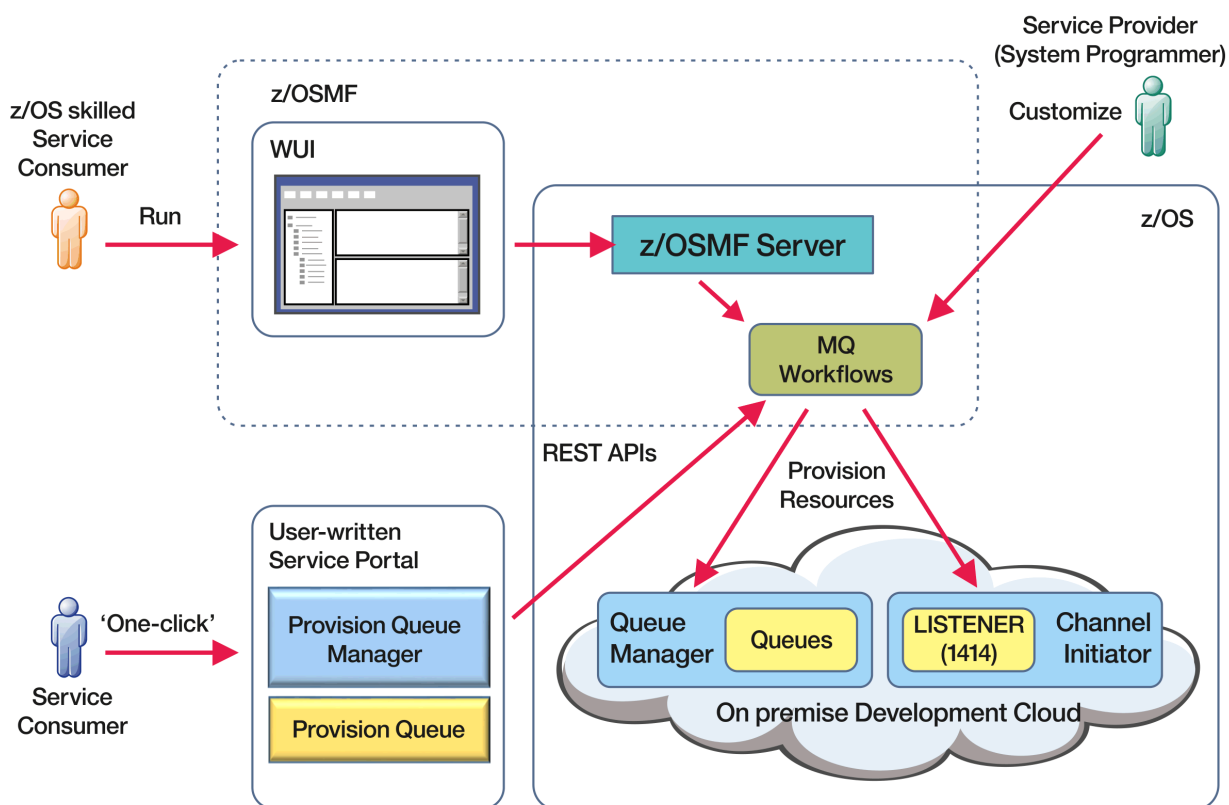
The following files declare variables that are referenced by the sample z/OSMF workflows and associated JCL or REXX exec files:

Workflow variable declaration file name	Description
common_variables.xml	Variables common to both the queue manager (plus channel initiator) and queue workflows.
qmgr_variables.xml	Variables specific to the queue manager (plus channel initiator) workflows.
queue_variables.xml	Variables specific to the queue workflows.
tcPIP_variables.xml	Variables specific to the queue manager (plus channel initiator) workflows, and used for identifying TCP/IP resources.

**Note:** The default visibility of variables is *private*. To allow variables to be queried using the z/OSMF REST API, selected variables have been marked as *public*. However, you can change the visibility of a given variable if required.

## Running the workflows

Figure 122. 'One-click' provisioning of IBM MQ for z/OS resources



Before the workflows can be run, some properties need to be set in the following file:

Workflow variable properties file name	Description
workflow_variables.properties	Initial properties for the workflow variables. Comments in the file indicate the purpose of each property. <ul style="list-style-type: none"> <li>Properties within meta-brackets (&lt; &gt;) need to be set to user specific values.</li> </ul>

Workflow variable properties file name	Description
	<ul style="list-style-type: none"> <li>An environment property can be set to provision queue managers for development (DEV), or test (TEST), or quality assurance (QA), or production (PROD) environments.</li> </ul> <p>Additional property settings control the characteristics of the queue manager to be provisioned for each environment. For example you can vary the number of active logs, or the number of page sets, for each environment type.</p> <ul style="list-style-type: none"> <li>Other properties are set to IBM MQ default values but can be modified to meet local conventions if required.</li> </ul>

In general, once the properties have been set, the workflows can be run as is. However, if required, you can customize a workflow to modify or remove existing steps, or to add new steps.

Workflows can be run:

- From the z/OSMF WUI.

From Cloud Provisioning -> Software Services in the WUI, workflows can be run in automatic or manual mode. The manual mode is useful when testing, and in both modes the progress of each step in the workflow can be monitored.

For more details, see [Cloud provisioning services](#) and [Create a workflow](#).

- Using the z/OSMF REST Workflow Services.

The REST Workflow Services can be used to run workflows through a REST API. This mode is useful for creating one-click operations from a user-written portal.

For more details, refer to [Cloud provisioning REST APIs](#), and also see [z/OSMF workflow services](#).

- Using the sample marketplace portal provided with z/OSMF.

### Related concepts

[“Automate the provisioning of IBM MQ objects” on page 1011](#)

Samples are supplied to automate the provisioning of queue managers and local queues.

## **Abilitazione degli agent MFT per connettersi a gestori code z/OS remoti**

In base alla titolarità, un agente MFT su z/OS può utilizzare una connessione client per connettersi a un gestore code z/OS . Ciò può portare a topologie IBM MQ più semplici.

Se un agent MFT su z/OS è associato all'identificativo del prodotto (PID) di IBM MQ Advanced for z/OS VUE o IBM MQ Advanced for z/OS, l'agent può utilizzare una connessione client per connettersi a un gestore code su z/OS.

Per informazioni sui diversi PID, consultare [IBM MQ product identifier and export information](#). 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.

Un agent MFT su z/OS, in esecuzione con qualsiasi altro PID, può connettersi solo a un gestore code locale utilizzando una connessione in modalità bind.

Se un agent tenta di connettersi a un gestore code non in esecuzione su z/OS, viene emesso il messaggio BFGQM1044E e l'avvio dell'agent viene terminato.

### Attività correlate

[Avvio di un agent MFT su z/OS](#)

[Arresto di un agent MFT su z/OS](#)



## Riferimenti correlati

[Il file MFT agent.properties](#)

# Configurazione di IBM MQ Internet Pass-Thru

---

Questa sezione descrive le varie funzioni supportate da IBM MQ Internet Pass-Thru (MQIPT) e come configurarle.

Configurare MQIPT apportando modifiche al file di configurazione `mqipt.conf`. La struttura del file di configurazione MQIPT e le proprietà che possono essere specificate sono descritte in [IBM MQ Internet Pass-Thru configuration reference](#).

**Nota:** È necessario impostare le autorizzazioni file protette sulla directory in cui si trova il file `mqipt.conf` per impedire agli utenti non autorizzati di visualizzare le password memorizzate o modificare la configurazione. Proteggere tutte le password specificate nel file di configurazione seguendo la procedura in [“Crittografia delle password memorizzate in MQIPT”](#) a pagina 1056.

Le modifiche al file di configurazione diventano effettive quando MQIPT viene avviato o aggiornato. L'aggiornamento di un'istanza attiva di MQIPT rende effettive le modifiche di configurazione senza riavviare MQIPT. Quando MQIPT viene aggiornato, il file di configurazione `mqipt.conf` viene riletto e MQIPT effettua quanto segue:



- Tutti gli instradamenti attivi contrassegnati come inattivi o non più specificati nel file di configurazione, vengono chiusi e non accettano più le connessioni in entrata.
- Tutti gli instradamenti contrassegnati come attivi nel file di configurazione e che non sono attualmente in esecuzione, vengono avviati.
- Vengono applicate tutte le modifiche ai parametri di configurazione degli instradamenti attivi. Laddove possibile, queste modifiche diventano effettive senza alcuna interruzione delle connessioni attive. Per alcune modifiche di parametri, come ad esempio una modifica alla destinazione dell'instradamento, tutte le connessioni vengono chiuse prima che la modifica venga applicata e che l'instradamento venga riavviato.

Per aggiornare MQIPT, utilizzare il comando `mqiptAdmin`. Per ulteriori informazioni sull'amministrazione di MQIPT mediante il comando `mqiptAdmin`, consultare [Amministrazione MQIPT utilizzando la riga comandi](#).

## Supporto per HTTP in MQIPT

MQIPT supporta il tunneling HTTP. MQIPT può essere configurato in modo che i pacchetti di dati inoltrati siano codificati come richieste HTTP.

I canali IBM MQ non accettano richieste HTTP. Pertanto, è necessario un secondo MQIPT per ricevere le richieste HTTP e convertirle nuovamente in pacchetti di protocollo IBM MQ. Il secondo MQIPT rimuove l'intestazione HTTP per convertire il pacchetto in entrata in un pacchetto di protocollo IBM MQ standard, prima di inoltrarlo al gestore code di destinazione.

**Nota:**   Da IBM MQ 9.4.0, gli instradamenti di MQIPT non accettano connessioni HTTP per impostazione predefinita. Le rotte devono essere configurate per accettare le connessioni HTTP utilizzando la proprietà `AllowedProtocols`.

Quando HTTP viene utilizzato tra due istanze di MQIPT, la connessione TCP/IP su cui il flusso di risposte e richieste HTTP è persistente e viene mantenuta aperta per la durata del canale di messaggi. MQIPT non chiude la connessione TCP/IP tra le coppie richiesta/risposta.

Se due istanze di MQIPT stanno comunicando tramite HTTP, è possibile che una richiesta HTTP rimanga in sospeso per un periodo prolungato. Un esempio è in un canale richiedente / server, quando il lato server è in attesa dell'arrivo di nuovi messaggi sulla propria coda di trasmissione. Il protocollo del canale IBM MQ fornisce un meccanismo "heartbeat", che richiede la fine dell'attesa periodicamente per inviare messaggi heartbeat al partner. Il periodo di heartbeat del canale predefinito è 5 minuti. MQIPT utilizza questo heartbeat come risposta HTTP. Non disabilitare questo heartbeat del canale o impostarlo su un valore eccessivamente elevato, per evitare di causare problemi con i timeout in alcuni firewall.

MQIPT accetta il traffico HTTP in formato suddiviso in blocchi, generato da un server o un proxy HTTP. Per un esempio di utilizzo di HTTP in MQIPT, consultare [Configurazione del tunneling HTTP](#).

## Proxy HTTP

Un proxy HTTP può essere inserito tra le due istanze di MQIPT. Il proxy HTTP deve soddisfare i seguenti requisiti:

- Il proxy deve supportare il protocollo HTTP 1.1 .
- Le intestazioni HTTP **Connection** o **Proxy-Connection** impostate da MQIPT devono essere rispettate dal proxy. Ciò consente di mantenere aperte le connessioni tra due istanze di MQIPT per la durata del canale di messaggi.
- È necessario mantenere un'associazione uno - a - uno di connessioni persistenti nel proxy. Ciò garantisce che le connessioni TCP/IP dal proxy alla destinazione MQIPT non vengano utilizzate per trasmettere i dati per più di un canale di messaggi.

È possibile impostare le proprietà per configurare la modalità di gestione delle connessioni persistenti su alcuni proxy HTTP. Ad esempio, è possibile impostare il numero massimo di richieste che possono essere effettuate su una connessione persistente. È necessario impostare le seguenti proprietà:

- Le connessioni persistenti dovrebbero essere abilitate.
- Il riutilizzo delle connessioni TCP/IP dal proxy a MQIPT da parte di più di una sessione HTTP deve essere disabilitato, per mantenere una mappatura uno - a - uno delle connessioni persistenti attraverso il proxy.
- Il timeout sulle richieste proxy deve essere impostata su un valore elevato. Ad esempio, 12 ore.
- Il numero massimo di richieste che possono essere effettuate su una connessione persistente deve essere impostato su un valore elevato. Ad esempio, 5000.

MQIPT utilizza le richieste HTTP POST per inviare i dati tra le due istanze di MQIPT. Se la configurazione di MQIPT specifica il nome host del proxy utilizzando la proprietà **HTTPProxy** , MQIPT si connette al proxy e utilizza il metodo HTTP CONNECT per richiedere che il proxy stabilisca un tunnel per la destinazione MQIPT. Ciò consente alle connessioni HTTPS di passare attraverso il proxy senza terminare la sessione TLS nel proxy.

Se un programma di bilanciamento del carico si trova tra le istanze MQIPT , deve essere configurato per utilizzare il valore del cookie HTTP *MQIPTSessionId* per garantire che tutte le richieste per ciascuna sessione vengano inoltrate alla stessa destinazione.

## HTTPS in MQIPT

HTTPS può essere utilizzato su una connessione HTTP abilitando le proprietà di instradamento **HTTPS** e **SSLClient** sul MQIPT che emette la connessione client.

MQIPT deve avere accesso al certificato CA attendibile che verrà utilizzato per autenticare il server / proxy HTTP di destinazione. La proprietà **SSLClientCAKeyring** può essere utilizzata per definire il file key ring contenente il certificato CA attendibile.

Una configurazione comune per HTTPS utilizzerà un proxy HTTP locale per eseguire il tunneling attraverso un firewall e connettersi a un server HTTP remoto (o un altro proxy), che a sua volta si connetterà al MQIPTremoto. Questo MQIPT sul lato server della connessione non richiede alcuna configurazione specifica, poiché la richiesta di connessione viene considerata come qualsiasi connessione HTTP normale.

MQIPT utilizza le proprietà **HTTPProxy** e **HTTPServer** per distinguere i proxy locali e remoti. L'instradamento MQIPT con la serie di proprietà **HTTPProxy** è visualizzato come proxy HTTP locale e l'instradamento MQIPT con la serie di proprietà **HTTPServer** è il server remoto (o proxy).

Di solito, le connessioni HTTPS vengono effettuate all'indirizzo della porta del listener 443 sul server / proxy HTTP, ma le proprietà **HTTPProxyPort** e **HTTPServerPort** possono essere utilizzate per sovrascrivere questo valore predefinito.

## Supporto SOCKS in MQIPT

Un proxy SOCKS è un servizio di rete utilizzato come punto di uscita controllato attraverso un firewall. Un'applicazione abilitata SOCKS, in esecuzione all'interno del firewall, può utilizzare il proxy SOCKS per connettersi a un'applicazione remota.

MQIPT può fungere da proxy SOCKS abilitando la propriet ... **SocksServer** , consentendo quindi a un'applicazione IBM MQ abilitata a SOCKS di connettersi tramite MQIPT a un gestore code IBM MQ remoto. Quando si utilizza questa funzione, la destinazione e l'indirizzo della porta di destinazione vengono ottenuti durante il processo di handshake SOCKS e, pertanto, le proprietà di instradamento **Destination** e **DestinationPort** vengono sovrascritte. Questa è una funzione chiave per il supporto del cluster IBM MQ .

MQIPT può anche agire come un client SOCKS, per conto di un'applicazione IBM MQ locale che non è stata abilitata SOCKS. Ciò è utile quando si utilizza un firewall che consente connessioni in uscita solo tramite un proxy SOCKS. Ogni instradamento MQIPT può essere configurato per comunicare con un proxy SOCKS differente.

Consultare [Configurazione di un proxy SOCKS](#) per un esempio di come utilizzare SOCKS.

## Cluster in MQIPT

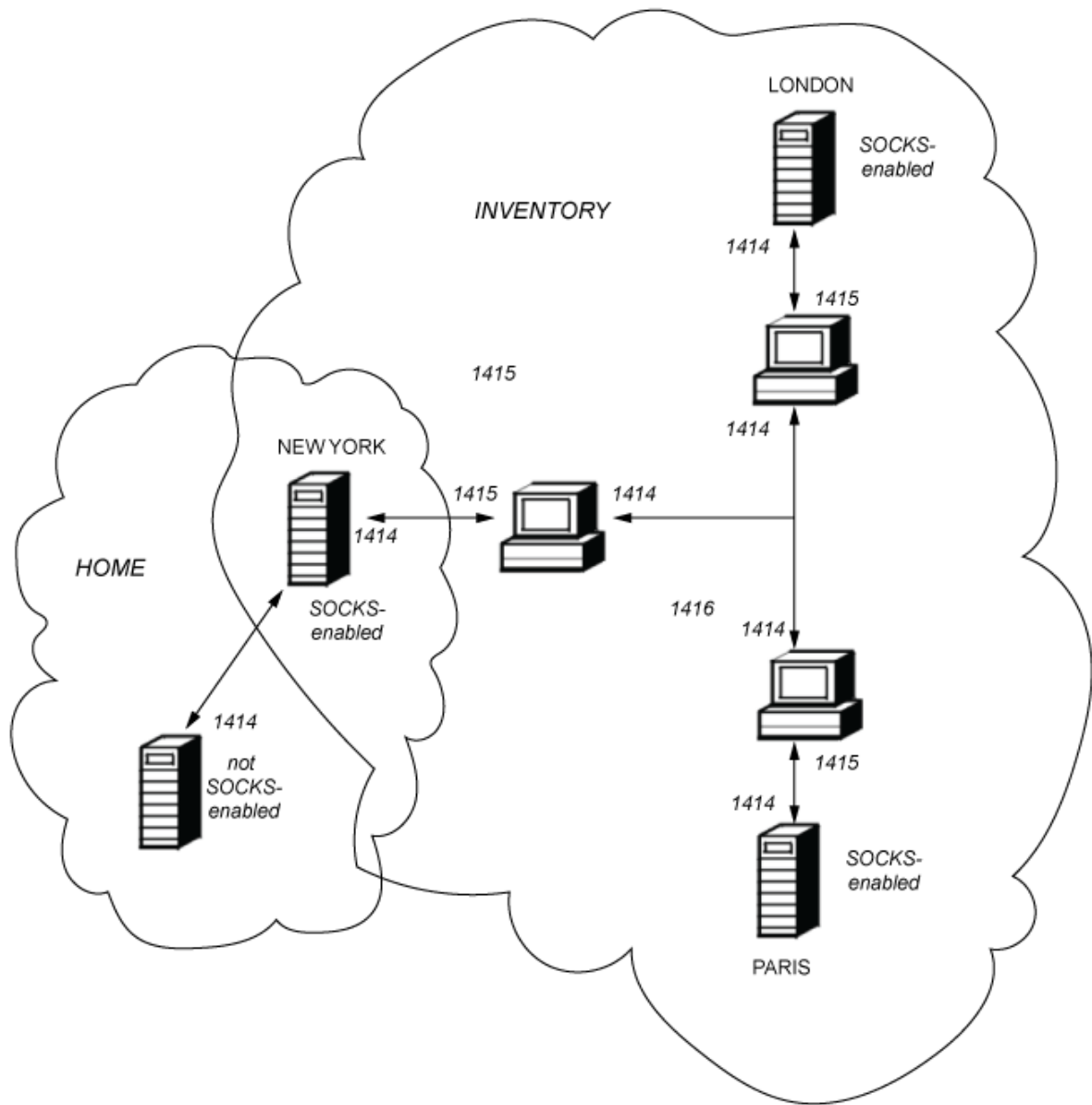
I cluster IBM MQ possono essere utilizzati con MQIPT da SOCKS - abilitando ogni gestore code nel cluster che si estende su Internet e abilitando MQIPT ad agire come proxy SOCKS.

Nel seguente diagramma, NEWYORK e CHICAGO si trovano in un cluster denominato HOME ed entrambi contengono repository completi. NEWYORK, LONDRA e PARIGI si trovano in un altro cluster chiamato INVENTORY. Nota che CHICAGO non ha bisogno di essere abilitato a SOCKS poiché si trova in un cluster che non ha bisogno di un MQIPT.

Ogni gestore code nel cluster INVENTORY è effettivamente "nascosto" dietro un MQIPT. Poiché il gestore code è stato abilitato a SOCKS, quando viene avviato un canale mittente del cluster, la richiesta viene inviata alla relativa destinazione, utilizzando MQIPT come proxy SOCKS. Di solito, il CONNAME su un canale ricevente del cluster viene usato per identificare il gestore code locale, ma quando viene usato con MQIPT, il CONNAME deve identificare il MQIPT locale e la sua porta listener in entrata. Nel seguente diagramma, tutti gli indirizzi della porta del listener in entrata sono 1414 e gli indirizzi della porta del listener in uscita sono 1415.

Esistono due metodi per eseguire un gestore code abilitato a SOCKS. Il primo è SOCKS - abilitare l'intero computer su cui è in esecuzione il gestore code. Il secondo è SOCKS - abilitare solo il gestore code. Utilizzando uno dei metodi, è necessario configurare il client SOCKS in modo che esegua solo connessioni remote utilizzando MQIPT come proxy SOCKS e disabilitare l'autenticazione utente. Ci sono una serie di prodotti sul mercato per ottenere il supporto SOCKS. È necessario sceglierne uno che supporti SOCKS V5 .

Consultare [Configurazione del supporto cluster MQIPT](#) per un esempio di configurazione di una rete cluster.



## Supporto SSL/TLS in MQIPT

I socket sicuri possono essere utilizzati per garantire la privacy delle comunicazioni, l'integrità delle comunicazioni e l'autenticazione.

### Riservatezza comunicazione

La connessione può essere resa privata. I dati da scambiare tra il client e il server possono essere codificati e solo il mittente e il destinatario possono dare un senso ai dati. Ciò significa che le informazioni private, come i numeri di carta di credito, possono essere trasferite in modo sicuro.

### Integrità della comunicazione

La connessione è affidabile. Il trasporto del messaggio include un controllo di integrità del messaggio basato su una funzione hash sicura.

### Autenticazione

Il client può autenticare il server e un server autenticato può autenticare il client. Ciò significa che le informazioni sono garantite per essere scambiate solo tra le parti previste. Il meccanismo di autenticazione si basa sullo scambio di certificati digitali (certificati X.509v3).

## Protocolli socket sicuri

In MQIPT, i socket sicuri vengono forniti utilizzando i protocolli TLS (Transport Layer Security) e SSL (Secure Sockets Layer). I due protocolli socket sicuri sono simili ma non interagiscono. In questa documentazione i termini SSL e TLS vengono utilizzati in modo intercambiabile a meno che non venga rilevata una differenza specifica.

MQIPT supporta SSL 3.0, TLS 1.0, TLS 1.1 e TLS 1.2 forniti da Java runtime environment (JRE). Da IBM MQ 9.3.0, MQIPT supporta anche TLS 1.3. La IBM MQ CipherSpec del canale remoto determina quale protocollo MQIPT utilizza.

SSL 3.0, TLS 1.0 e TLS 1.1 non sono sicuri e sono disabilitati per impostazione predefinita in MQIPT. Se è necessario utilizzare uno di questi protocolli disabilitati, è possibile riabilitarli seguendo la procedura in [“Abilitazione di protocolli obsoleti e suite di crittografia in MQIPT”](#) a pagina 1044.

I protocolli SSL/TLS possono utilizzare diversi algoritmi di firma digitale per l'autenticazione delle parti di comunicazione. Le operazioni di crittografia utilizzate in SSL/TLS, la crittografia per la riservatezza dei dati e l'hashing sicuro per l'integrità dei messaggi, si basano sulla condivisione delle chiavi segrete tra client e server. SSL/TLS fornisce vari meccanismi di scambio di chiavi che consentono la condivisione di chiavi segrete. SSL/TLS può utilizzare vari algoritmi per la crittografia e l'hashing.

## Abilitazione della modalità FIPS in MQIPT

Il componente crittografico SSL/TLS del JRE contiene il provider di sicurezza IBMJCEPlusFIPS, che è certificato conforme allo standard FIPS 140-2. Se si desidera utilizzare solo la crittografia certificata FIPS in MQIPT, abilitare la modalità FIPS nel provider IBMJSSE2 impostando le seguenti Java proprietà di sistema all'avvio di MQIPT:

- `com.ibm.jsse2.usefipsprovider=true`
- `com.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS`

È possibile impostare le Java proprietà di sistema quando MQIPT viene avviato utilizzando la variabile di ambiente **MQIPT\_JVM\_OPTIONS**. Ad esempio, in Linux, immettere il seguente comando per impostare la variabile di ambiente, prima di immettere il comando per avviare MQIPT:

```
export MQIPT_JVM_OPTIONS="-Dcom.ibm.jsse2.usefipsprovider=true
-Dcom.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS"
```

Per ulteriori informazioni sull'abilitazione della modalità FIPS, consultare [Abilitazione della modalità FIPS nel provider IBMJSSE2](#).

## Modalità bridging SSL/TLS

Quando un instradamento ha sia SSLServer che SSLClient impostati, il MQIPT accetta una connessione protetta SSL/TLS in entrata e stabilisce una seconda connessione protetta SSL/TLS a un altro MQIPT o a un gestore code di destinazione. Le informazioni sul canale IBM MQ vengono decodificate e ricodificate tra queste due connessioni SSL/TLS. Il bridge SSL/TLS è anche indicato come *proxy di terminazione SSL/TLS*.

IBM MQ supporta il bridging SSL/TLS utilizzando MQIPT. Altri proxy di terminazione SSL/TLS con IBM MQ sono stati osservati per causare connessioni interrotte se il proxy combina o ricostruisce i record SSL/TLS con dimensioni diverse rispetto a quelle inviate da IBM MQ. Ciò è dovuto ad un'interazione tra il modo in cui i gestori code assegnano e gestiscono la memoria per i dati di rete IBM MQ in entrata e il modo in cui i dati di rete IBM MQ vengono compressi nei record SSL/TLS.

MQIPT conserva il packaging dei dati di rete IBM MQ nei record SSL/TLS senza suddividerli o combinarli. Se altri bridge SSL/TLS non conservano esattamente i record SSL/TLS, possono causare errori dei canali IBM MQ con messaggi di errore:

```
AMQ9638: SSL communications error for channel
AMQ9208: Error on receive from host
```

## Modalità proxy SSL/TLS

Una rotta MQIPT può essere configurata in modalità proxy SSL/TLS come alternativa al bridging SSL/TLS. In questa modalità, l'instradamento inoltra solo i dati SSL/TLS tra due endpoint IBM MQ ; non partecipa all'handshake SSL/TLS e non richiede alcun certificato digitale.

È possibile utilizzare la modalità proxy SSL/TLS nei casi in cui i canali IBM MQ che comunicano tramite MQIPT sono già configurati per la comunicazione SSL/TLS e si desidera utilizzare MQIPT per un altro scopo, ad esempio l'instradamento di connessioni tramite firewall o la limitazione della serie di connessioni consentite tramite un'uscita di sicurezza. Durante l'esecuzione in modalità proxy SSL/TLS, MQIPT verifica che i pacchetti SSL/TLS iniziali ricevuti da una nuova connessione siano validi prima di inoltrare i pacchetti alla destinazione.

IBM MQ supporta la modalità proxy SSL/TLS con MQIPT o qualsiasi altro proxy SSL/TLS

## IBM MQ supporto per più certificati con MQIPT

IBM MQ 8.0e versioni successive, supporta l'utilizzo di più certificati sullo stesso gestore code, utilizzando un'etichetta di certificato per canale, specificata utilizzando l'attributo **CERTLABL** nella definizione del canale. I canali in entrata per il gestore code (ad esempio, la connessione server o il destinatario) si basano sul rilevamento del nome del canale utilizzando SNI (Server Name Indication) TLS, al fine di presentare il certificato corretto dal gestore code. Per ulteriori informazioni sull'utilizzo di più certificati su un gestore code, fare riferimento a [Come IBM MQ fornisce più capacità di certificati](#).

Se un canale si connette al gestore code di destinazione tramite MQIPT e l'instradamento MQIPT ha sia **SSLServer** che **SSLClient** impostati, ci sono due sessioni TLS separate tra gli endpoint. Nelle versioni precedenti a IBM MQ 9.3.0, i dati SNI non passano attraverso l'interruzione della sessione. Ciò impedisce che un certificato per canale venga utilizzato sul gestore code di destinazione, per la connessione TLS tra MQIPT e il gestore code. Per utilizzare un certificato per canale sul gestore code di destinazione, per una connessione TLS che passa attraverso MQIPT a una versione precedente a IBM MQ 9.3.0, l'instradamento MQIPT deve utilizzare la modalità proxy SSL/TLS, che inoltra tutti i flussi di controllo TLS intatti, incluso il nome SNI.

Da IBM MQ 9.3.0, MQIPT può essere configurato utilizzando la proprietà di instradamento `SSLClientOutboundSNI` per impostare l'SNI per le connessioni TLS su un valore specifico o per passare attraverso l'SNI ricevuto sulla connessione in entrata all'instradamento. Per consentire ai certificati per canale di essere utilizzati su un gestore code di destinazione, l'instradamento deve essere configurato per impostare l'SNI sul nome canale IBM MQ o per passare attraverso l'SNI ricevuto sulla connessione in entrata all'instradamento. Se MQIPT è configurato per passare attraverso l'SNI, il gestore code o il client che si connette a MQIPT deve impostare l'SNI sul nome del canale.

I certificati utilizzati per le connessioni TLS terminate o avviate da MQIPT possono essere configurati singolarmente per ogni instradamento, ad esempio utilizzando le proprietà di instradamento `SSLServerSiteLabel` e `SSLClientSiteLabel`.

## CipherSuites supportati da MQIPT

Tabella 71 a pagina 1023 mostra quali CipherSuites sono supportati da MQIPT e quali sono abilitati per impostazione predefinita.

Per impostazione predefinita, è abilitata solo una sottoserie di CipherSuites . CipherSuites basato su diversi algoritmi considerati non sicuri sono disabilitati da JRE. Se si è consapevoli dei potenziali rischi, ma è ancora necessario utilizzare uno di questi CipherSuites, è possibile aggiungere il supporto per una CipherSuite disabilitata seguendo la procedura riportata in [“Abilitazione di protocolli obsoleti e suite di crittografia in MQIPT”](#) a pagina 1044.

Tabella 71. CipherSuites che è possibile utilizzare con MQIPT

CipherSuite	Abilitato per impostazione predefinita.
<b>CipherSuites per TLS 1.3</b>	
TLS_AES_128_GCM_SHA256	Sì
TLS_AES_256_GCM_SHA384	Sì
TLS_CHACHA20_POLY1305_SHA256	Sì
<b>CipherSuites per SSL 3.0, TLS 1.0, TLS 1.1 e TLS 1.2</b>	
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA256	
SSL_DH_anon_WITH_AES_128_GCM_SHA256	
SSL_DH_anon_WITH_AES_256_CBC_SHA	
SSL_DH_anon_WITH_AES_256_CBC_SHA256	
SSL_DH_anon_WITH_AES_256_GCM_SHA384	
SSL_DH_anonimo_WITH_DES_CBC_SHA	
SSL_DH_anon_WITH_RC4_128_MD5	
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_DSS_WITH_AES_128_CBC_SHA	Sì
SSL_DHE_DSS_WITH_AES_128_CBC_SHA256	Sì
SSL_DHE_DSS_WITH_AES_128_GCM_SHA256	Sì
SSL_DHE_DSS_WITH_AES_256_CBC_SHA	Sì
SSL_DHE_DSS_WITH_AES_256_CBC_SHA256	Sì
SSL_DHE_DSS_WITH_AES_256_GCM_SHA384	Sì
SSL_DHE_DSS_WITH_DES_CBC_SHA	
SSL_DHE_DSS_WITH_RC4_128_SHA	
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_RSA_WITH_AES_128_CBC_SHA	Sì
SSL_DHE_RSA_WITH_AES_128_CBC_SHA256	Sì
SSL_DHE_RSA_WITH_AES_128_GCM_SHA256	Sì
SSL_DHE_RSA_WITH_AES_256_CBC_SHA	Sì
SSL_DHE_RSA_WITH_AES_256_CBC_SHA256	Sì

Tabella 71. CipherSuites che è possibile utilizzare con MQIPT (Continua)

CipherSuite	Abilitato per impostazione predefinita.
SSL_DHE_RSA_WITH_AES_256_GCM_SHA384	Sì
SSL_DHE_RSA_WITH_DES_CBC_SHA	
SSL_ECDH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_anon_WITH_AES_128_CBC_SHA	
SSL_ECDH_anon_WITH_AES_256_CBC_SHA	
SSL_ECDH_anon_WITH_NULL_SHA	
SSL_ECDH_anon_WITH_RC4_128_SHA	
SSL_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA	Sì
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	Sì
SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	Sì
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA	Sì
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	Sì
SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	Sì
SSL_ECDH_ECDSA_WITH_NULL_SHA	
SSL_ECDH_ECDSA_WITH_RC4_128_SHA	
SSL_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA	Sì
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256	Sì
SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256	Sì
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA	Sì
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384	Sì
SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384	Sì
SSL_ECDH_RSA_WITH_NULL_SHA	
SSL_ECDH_RSA_WITH_RC4_128_SHA	
SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	Sì
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	Sì
SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	Sì
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	Sì
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	Sì
SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	Sì
SSL_ECDHE_ECDSA_WITH_NULL_SHA	



Tabella 71. CipherSuites che è possibile utilizzare con MQIPT (Continua)

CipherSuite	Abilitato per impostazione predefinita.
SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	
SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA	Sì
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	Sì
SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	Sì
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA	Sì
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	Sì
SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Sì
SSL_ECDHE_RSA_WITH_NULL_SHA	
SSL_ECDHE_RSA_WITH_RC4_128_SHA	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA	
SSL_KRB5_EXPORT_WITH_RC4_40_MD5	
SSL_KRB5_EXPORT_WITH_RC4_40_SHA	
SSL_KRB5_WITH_3DES_EDE_CBC_MD5	
SSL_KRB5_WITH_3DES_EDE_CBC_SHA	
SSL_KRB5_WITH_DES_CBC_MD5	
SSL_KRB5_WITH_DES_CBC_SHA	
SSL_KRB5_WITH_RC4_128_MD5	
SSL_KRB5_WITH_RC4_128_SHA	
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_RSA_EXPORT_WITH_RC4_40_MD5	
SSL_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_RSA_WITH_AES_128_CBC_SHA	Sì
SSL_RSA_WITH_AES_128_CBC_SHA256	Sì
SSL_RSA_WITH_AES_128_GCM_SHA256	Sì
SSL_RSA_WITH_AES_256_CBC_SHA	Sì
SSL_RSA_WITH_AES_256_CBC_SHA256	Sì
SSL_RSA_WITH_AES_256_GCM_SHA384	Sì
SSL_RSA_WITH_DES_CBC_SHA	
SSL_RSA_WITH_NULL_MD5	
SSL_RSA_WITH_NULL_SHA	
SSL_RSA_WITH_NULL_SHA256	

Tabella 71. CipherSuites che è possibile utilizzare con MQIPT (Continua)	
CipherSuite	Abilitato per impostazione predefinita.
SSL_RSA_WITH_RC4_128_MD5	Sì
SSL_RSA_WITH_RC4_128_SHA	
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	

## CipherSpecs e MQIPT CipherSuites

Tabella 72 a pagina 1027 mostra la relazione tra le CipherSpecs supportate da IBM MQ e le CipherSuites supportate da MQIPT. La tabella mostra anche la versione del protocollo che IBM MQ si aspetta che ogni CipherSpec utilizzi.

Quando un gestore code o un client IBM MQ comunica con MQIPT, a meno che non si stia utilizzando la modalità proxy SSL in MQIPT, assicurarsi che CipherSpec utilizzato da IBM MQ corrisponda a CipherSuite utilizzato da MQIPT:

- Quando MQIPT agisce come server TLS e IBM MQ si connette come client TLS, il CipherSpec utilizzato da IBM MQ deve corrispondere a una CipherSuite selezionata nella configurazione della rotta MQIPT .
- Quando MQIPT funge da client TLS e si connette a un gestore code IBM MQ che funge da server TLS, MQIPT CipherSuite deve corrispondere alla CipherSpec definita sul canale di ricezione IBM MQ .

Una IBM MQ CipherSpec determina in maniera univoca sia l'algoritmo di codifica che la versione del protocollo socket sicuro da utilizzare. Alcuni IBM MQ CipherSpecs differiscono solo per versione del protocollo. Se si utilizza uno di questi CipherSpecs , non è sufficiente specificare solo la CipherSuite nella configurazione MQIPT . L'handshake SSL/TLS negozia la versione più elevata del protocollo dei socket sicuri supportata da entrambi i lati e seleziona quindi una CipherSuite dalla serie di cifrature reciprocamente abilitate.

Ad esempio, un instradamento SSLClient con `SSLClientCipherSuites=SSL_RSA_WITH_3DES_EDE_CBC_SHA` può negoziare l'utilizzo di `TLS_RSA_WITH_3DES_EDE_CBC_SHA` (TLS 1.0) o `TRIPLE_DES_SHA_US` (SSL 3.0) con il gestore code remoto. È anche possibile utilizzare questa CipherSuite con TLS 1.2, ma IBM MQ non supporta questa CipherSuite con TLS 1.2. Per questo motivo, è particolarmente probabile che gli instradamenti SSLClient causino errori AMQ9616 o AMQ9631 nel gestore code.

Per evitare tali errori sugli instradamenti SSLClient, impostare la proprietà di instradamento **SSLClientProtocols** sul valore appropriato per la CipherSpec desiderata. In alcuni casi potrebbe essere necessario limitare la serie di protocolli lato server utilizzando la proprietà di instradamento **SSLServerProtocols** . Utilizzare la versione del protocollo mostrata nella tabella per determinare l'impostazione corretta per queste proprietà di instradamento.

Questo problema riguarda in particolare le seguenti CipherSuites e CipherSpecs per le rotte SSLClient:

- `SSL_RSA_WITH_3DES_EDE_CBC_SHA`, che corrisponde a:
  - SSL 3.0: MQ CipherSpec `TRIPLE_DES_SHA_US`
  - TLS 1.0: MQ CipherSpec `TLS_RSA_WITH_3DES_EDE_CBC_SHA`
- `SSL_RSA_WITH_DES_CBC_SHA`, che corrisponde a:
  - SSL 3.0: MQ CipherSpec `DES_SHA_EXPORT`
  - TLS 1.0: MQ CipherSpec `TLS_RSA_WITH_DES_CBC_SHA`
- `SSL_RSA_WITH_RC4_128_SHA`, che corrisponde a:

- SSL 3.0: MQ CipherSpec RC4\_SHA\_US
- TLS 1.2: MQ CipherSpec TLS\_RSA\_WITH\_RC4\_128\_SHA256

Se si utilizza un singolo instradamento MQIPT SSLClient per eseguire il tunnel di diversi canali IBM MQ che utilizzano CipherSpecs differenti, assicurarsi che tutti i canali siano configurati con un CipherSpec che utilizza la stessa versione del protocollo SSL e che la proprietà di instradamento **SSLClientProtocols** sia impostata su questo protocollo.

Per ulteriori informazioni su IBM MQ CipherSpecs, consultare Abilitazione di CipherSpecs.

*Tabella 72. IBM MQ CipherSpecs che corrispondono a MQIPT CipherSuites*

<b>IBM MQ CipherSpec</b>	<b>MQIPT CipherSuite</b>	<b>Version e protocollo</b>
DES_SHA_EXPORT	SSL_RSA_WITH_DES_CBC_SHA	SSLv3
DES_SHA_EXPORT1024	N/A	N/A
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_ECDSA_AES_128_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_ECDSA_AES_128_GCM_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_ECDSA_AES_256_CBC_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_ECDSA_AES_256_GCM_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_ECDSA_NULL_SHA256	SSL_ECDHE_ECDSA_WITH_NULL_SHA	TLSv1.2
ECDHE_ECDSA_RC4_128_SHA256	SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	TLSv1.2
ECDHE_RSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_RSA_AES_128_CBC_SHA256	SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_RSA_AES_128_GCM_SHA256	SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_RSA_AES_256_CBC_SHA384	SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_RSA_AES_256_GCM_SHA384	SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_RSA_NULL_SHA256	SSL_ECDHE_RSA_WITH_NULL_SHA	TLSv1.2
ECDHE_RSA_RC4_128_SHA256	SSL_ECDHE_RSA_WITH_RC4_128_SHA	TLSv1.2
NULL_MD5	SSL_RSA_WITH_NULL_MD5	SSLv3
NULL_SHA	SSL_RSA_WITH_NULL_SHA	SSLv3
RC2_MD5_EXPORT	N/A	N/A
RC4_56_SHA_EXPORT1024	N/A	N/A
RC4_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSLv3
RC4_MD5_US	SSL_RSA_WITH_RC4_128_MD5	SSLv3
RC4_SHA_US	SSL_RSA_WITH_RC4_128_SHA	SSLv3
TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256	TLSv1.3

Tabella 72. IBM MQ CipherSpecs che corrispondono a MQIPT CipherSuites (Continua)

IBM MQ CipherSpec	MQIPT CipherSuite	Version e protocollo
TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384	TLSv1.3
TLS_CHACHA20_POLY1305_SHA256	TLS_CHACHA20_POLY1305_SHA256	TLSv1.3
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA256	SSL_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_128_GCM_SHA256	SSL_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA256	SSL_RSA_WITH_AES_256_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_GCM_SHA384	SSL_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_NULL_NULL	N/A	N/A
TLS_RSA_WITH_NULL_SHA256	SSL_RSA_WITH_NULL_SHA256	TLSv1.2
TLS_RSA_WITH_RC4_128_SHA256	SSL_RSA_WITH_RC4_128_SHA	TLSv1.2
TRIPLE_DES_SHA_US	SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSLv3

## Handshake SSL/TLS in MQIPT

Il processo di handshake SSL/TLS si verifica durante la richiesta di connessione iniziale tra il server e il client SSL/TLS, quando viene eseguita l'autenticazione e la negoziazione di CipherSuites .

Tutte le CipherSuites SSL/TLS supportate (vedere “Supporto SSL/TLS in MQIPT” a pagina 1020), ad eccezione delle CipherSuites anonime, richiedono l'autenticazione server e consentono l'autenticazione client; il server può essere configurato per richiedere l'autenticazione client. Dovresti evitare di usare CipherSuites anonime perché non forniscono alcuna garanzia sull'identità del peer remoto. È possibile per un attacco man - in - the - middle intercettare connessioni SSL/TLS anonime a tua insaputa. Utilizzare CipherSuites anonime solo su reti interne affidabili e solo se si è pronti ad accettare il rischio di intercettazione dei dati.

L'autenticazione peer di comunicazione in SSL/TLS si basa sulla codifica della chiave pubblica e sui certificati digitali X.509v3 . Un sito che deve essere autenticato nel protocollo SSL/TLS richiede una chiave privata e un certificato digitale (che contiene la chiave pubblica corrispondente insieme alle informazioni sull'identità del sito), tempo di validità del certificato. I certificati sono firmati da un'autorità di certificazione, i certificati di tali autorità sono chiamati certificati del firmatario. Un certificato seguito da uno o più certificati del firmatario costituisce una catena di certificati. Una catena di certificati è caratterizzata dal fatto che, a partire dal primo certificato (certificato del sito), la firma di ogni certificato nella catena può essere verificata utilizzando la chiave pubblica contenuta nel certificato del firmatario successivo.

Quando viene stabilita una connessione sicura che richiede l'autenticazione del server, il server invia al client una catena di certificati per provare la sua identità. Il client SSL/TLS continuerà a stabilire la connessione al server solo se è in grado di autenticare il server, ad esempio, verificare la firma del certificato del sito del server. Per verificare la firma, il client SSL/TLS deve considerare attendibile il sito del server stesso o almeno uno dei firmatari nella catena di certificati fornita dal server. I certificati dei siti attendibili e dei firmatari devono essere conservati sul lato client per eseguire questa verifica.

Il client SSL/TLS ispeziona la catena di certificati del server, iniziando con il certificato del sito. Il client considera valida la firma del certificato del sito nelle seguenti circostanze:

- Il certificato del sito si trova nel repository del sito attendibile o dei certificati del firmatario
- Un certificato del firmatario nella catena può essere convalidato in base al relativo repository di certificati del firmatario attendibili

In quest' ultimo caso, il client SSL/TLS controlla che la catena di certificati sia effettivamente firmata correttamente, dal certificato del firmatario attendibile fino al certificato del sito del server. Ogni certificato coinvolto in questo processo viene anche esaminato per la correttezza del formato e delle date di validità. Se uno di questi controlli non riesce, la connessione al server viene rifiutata. Dopo aver verificato il certificato server, il client utilizza la chiave pubblica incorporata in tale certificato nei passi successivi del protocollo SSL/TLS. La connessione SSL/TLS può essere stabilita solo se il server ha realmente la chiave privata corrispondente.

L'autenticazione client segue la stessa procedura: se un server SSL/TLS richiede l'autenticazione client, il client invia al server una catena di certificati per provare la sua identità. Il server verifica la catena in base al relativo repository di certificati del firmatario e del sito attendibile. Dopo aver verificato il certificato del client, il server utilizza la chiave pubblica incorporata in tale certificato nei passi successivi del protocollo SSL/TLS. La connessione SSL/TLS può essere stabilita solo se il client ha realmente la chiave privata corrispondente.

Le versioni recenti dei protocolli TLS forniscono comunicazioni di alta sicurezza (SSL e i protocolli TLS precedenti sono considerati non sicuri). Tuttavia, il protocollo funziona in base alle informazioni fornite dall'applicazione. Solo se tale base di informazioni viene mantenuta in modo sicuro, è possibile raggiungere l'obiettivo generale di una comunicazione sicura. Ad esempio, se il repository dei certificati del sito attendibile e del firmatario è compromesso, è possibile stabilire una connessione sicura a un partner di comunicazione molto non sicuro.

## Implementazione MQIPT di SSL/TLS

SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2 e TLS 1.3 sono implementati con token PKCS (Public Key Cryptography Standards) #12 memorizzati in file keyring (con tipi di file .p12 o .pfx), contenenti X509.V3 V3. MQIPT può anche utilizzare keystore hardware crittografici che supportano lo standard PKCS#11 Cryptographic Token Interface. MQIPT utilizza il package JSSE ( IBM Java Secure Socket Extension ).

MQIPT può fungere da client SSL/TLS o da server SSL/TLS a seconda di quale estremità avvia la connessione. Il client avvia una connessione e il server accetta la richiesta di connessione. È possibile che un instradamento MQIPT agisca sia come client che come server. In questo caso, l'utilizzo della funzione Modalità proxy SSL/TLS in genere fornisce prestazioni migliori.

Quando MQIPT è configurato per la modalità proxy SSL/TLS, inoltra solo i dati SSL/TLS tra i due endpoint; non partecipa all'handshake SSL/TLS e non richiede alcun certificato digitale.

Nelle versioni precedenti a IBM MQ 9.3.0, MQIPT non trasmette i dati SNI (Server Name Indication) TLS ricevuti su una connessione TLS in entrata tramite una connessione TLS in uscita. Ciò significa che i certificati per canale, specificati utilizzando l'attributo del canale **CERTLABL** , non possono essere utilizzati per le connessioni TLS tra MQIPT e il gestore code di destinazione. Per utilizzare un certificato per canale sul gestore code di destinazione, per una connessione TLS che passa attraverso MQIPT a una versione precedente a IBM MQ 9.3.0, l'instradamento MQIPT deve utilizzare la modalità proxy SSL/TLS, che inoltra tutti i flussi di controllo TLS intatti, incluso il nome SNI. Da IBM MQ 9.3.0, MQIPT può essere configurato per impostare l'SNI per connessioni TLS su un valore specifico o per passare attraverso l'SNI ricevuto sulla connessione in entrata all'instradamento. Per ulteriori informazioni sull'utilizzo di più certificati su un gestore code con MQIPT, consultare [“IBM MQ supporto per più certificati con MQIPT” a pagina 1022](#).

Ogni instradamento MQIPT può essere configurato in maniera indipendente con la propria serie di proprietà SSL/TLS. Per ulteriori dettagli, consultare [MQIPT proprietà di instradamento](#) .

## Codifica di una password key ring in MQIPT

Le parole d'ordine utilizzate da MQIPT per aprire un file key ring o per accedere all'hardware crittografico, devono essere codificate con il comando **mqiptPW**.

La password codificata può essere specificata in una delle seguenti proprietà:

- **SSLClientKeyRingPW**
- **SSLClientCAKeyRingPW**
- **SSLServerKeyRingPW**
- **SSLServerCAKeyRingPW**
- **SSLCommandPortKeyRingPW**

Nelle versioni precedenti a IBM MQ 9.1.5, le password keyring per l'utilizzo da MQIPT sono memorizzate nei file a cui fanno riferimento le proprietà **SSL\*KeyRingPW**.

Codificare le password keyring per l'utilizzo da parte di MQIPT utilizzando il comando **mqiptPW** e impostare il valore delle proprietà **SSL\*KeyRingPW** sulla password codificata. MQIPT può distinguere tra password codificate e nomi file nei valori delle proprietà per la compatibilità con le configurazioni create prima di IBM MQ 9.1.5.

**Deprecated** Il metodo di codifica delle password key ring nelle versioni MQIPT precedenti a IBM MQ 9.1.5 è obsoleto, ma può essere ancora utilizzato. Per migliorare la protezione delle password key ring, codificare nuovamente le password key ring codificate utilizzando il metodo obsoleto, con il metodo di protezione più recente.

Per codificare una password key ring per l'utilizzo da parte di MQIPT, seguire la procedura in [“Crittografia delle password memorizzate in MQIPT”](#) a pagina 1056.

## Selezione di certificati da un file di chiavi in MQIPT

È possibile avere più di un certificato personale memorizzato nello stesso file key ring o token hardware crittografico. Le proprietà **SSLClientSite\*** possono essere utilizzate sul lato client per selezionare il certificato da inviare al server per l'autenticazione e le proprietà **SSLServerSite\*** possono essere utilizzate sul lato server per selezionare il certificato da inviare al client per l'autenticazione.

Utilizzando queste proprietà, è possibile selezionare un certificato in base al relativo DN (Distinguished Name). In alternativa, l'etichetta del certificato può essere utilizzata per selezionare un certificato utilizzando la proprietà **SSLServerSiteLabel** e **SSLClientSiteLabel**.

Per selezionare il certificato server utilizzato dalla porta comandi TLS, utilizzare la proprietà **SSLCommandPortSiteLabel** per specificare il nome etichetta del certificato.

## Impostazioni di attendibilità in MQIPT

Un archivio chiavi contiene un certificato personale che include il certificato del firmatario o la catena di certificati del firmatario.

MQIPT utilizza due tipi di keystore:

### Keystore CA (Certificate Authority)

Questo keystore contiene i certificati CA attendibili utilizzati per convalidare i certificati che appartengono a un peer remoto. Questi certificati CA aiutano a determinare se il peer remoto è affidabile. MQIPT supporta keystore in formato PKCS #12 e keystore hardware crittografico che supportano l'interfaccia PKCS #11, per la memorizzazione dei certificati CA.

I keystore CA di MQIPT sono identificati dalle proprietà di instradamento **SSLClientCAKeyRing** e **SSLServerCAKeyRing**. L'utilizzo dell'hardware crittografico per accedere ai certificati CA è abilitato impostando la proprietà **SSLClientCAKeyRingUseCryptoHardware** e **SSLServerCAKeyRingUseCryptoHardware**.

Il keystore CA sul lato client SSL/TLS deve contenere i certificati CA attendibili utilizzati per autenticare il certificato inviato dal server. Se un instradamento del server SSL è configurato per

l'autenticazione client, il keystore CA sul lato server SSL/TLS deve contenere i certificati CA attendibili utilizzati per autenticare il certificato inviato dal client.

### Keystore certificato personale

Questo keystore contiene i certificati personali che MQIPT utilizza per identificarsi con un peer remoto. Quando si genera un certificato autofirmato o si richiede un certificato firmato da una CA, farlo utilizzando il keystore del certificato personale.

MQIPT supporta sia i keystore in formato PKCS #12 che i keystore hardware crittografici che supportano l'interfaccia PKCS #11, per memorizzare certificati personali.

I keystore del certificato personale sono identificati dalle proprietà di instradamento **SSLClientKeyRing** e **SSLServerKeyRing**. L'utilizzo dell'hardware crittografico per accedere ai certificati personali è abilitato impostando la proprietà **SSLClientKeyRingUseCryptoHardware** e **SSLServerKeyRingUseCryptoHardware**.


Il keystore sul lato server SSL/TLS deve contenere il certificato personale del server MQIPT. Se l'autenticazione client è necessaria su un instradamento client SSL, il keystore sul lato client SSL/TLS deve contenere il certificato personale del client.

Se è necessaria l'autenticazione client, è necessario abilitare la proprietà **SSLServerAskClientAuth** sul lato server. Il keystore sul lato client deve contenere il certificato personale del client. Il keystore MQIPT sul lato server, identificato dalla proprietà **SSLServerCAKeyRing**, deve contenere certificati CA attendibili utilizzati per autenticare il client.

Se non si configura un keystore CA per un instradamento, MQIPT ricerca i certificati CA nel keystore del certificato personale, se ne è configurato uno. Ad esempio, se non è impostato alcun valore per **SSLServerCAKeyRing**, MQIPT ricerca i certificati CA nel keystore identificato da **SSLServerKeyRing**.

Come alternativa all'uso di certificati firmati da una CA attendibile, è possibile utilizzare certificati autofirmati. È possibile trovare un esempio di certificato autofirmato nel keystore di esempio `sslSample.pfx` fornito con MQIPT nella directory secondaria `samples/ssl`. Per aprire i keystore PKCS#12 di esempio, è necessario utilizzare la password `mqiptSample`.

I certificati autofirmati possono essere utili negli scenari di test in cui è necessario garantire la connettività SSL/TLS senza pagare una CA per un certificato. Tuttavia, non utilizzare certificati autofirmati negli ambienti di produzione. Per creare un certificato firmato dalla CA, consultare [Creazione di un file key ring](#).

È possibile utilizzare il comando  **mqiptKeytool** per gestire i keystore e i certificati digitali. Per ulteriori informazioni, consultare [“Gestione dei keystore MQIPT” a pagina 1035](#).

Proteggere i keystore e i file di password utilizzando le funzioni di sicurezza del sistema operativo per impedire l'accesso non autorizzato a tali file.

## Test di SSL/TLS in MQIPT

Esempi per aiutarti a testare una connessione SSL/TLS.

Per una descrizione dei vari scenari, vedi [Introduzione a IBM MQ Internet Pass-Thru](#). In particolare, consultare le seguenti attività:

- [Autenticazione di un server SSL/TLS](#)
- [Autenticazione di un client SSL/TLS](#)
- [Esecuzione MQIPT in modalità proxy SSL/TLS](#)
- [Esecuzione MQIPT in modalità proxy SSL/TLS con un gestore sicurezza](#)

Per verificare che la configurazione SSL/TLS funzioni correttamente, è possibile utilizzare i certificati autofirmati. I certificati autofirmati sono utili negli scenari di test in modo da poter verificare la connettività SSL/TLS senza pagare una CA (Certificate Authority) per un certificato. Per ulteriori informazioni, consultare [Creazione di certificati di test](#).

Non utilizzare alcun certificato autofirmato negli ambienti di produzione. Invece, ottenere un certificato firmato da una CA attendibile. Per creare un certificato firmato dalla CA, consultare [Creazione di un file key ring](#).

Quando si crea o si richiede un certificato, è necessario considerare quale tipo di chiave, dimensione chiave e algoritmo di firma digitale sono appropriati per le proprie necessità di sicurezza. Per ulteriori informazioni, consultare [“Considerazioni sul certificato digitale per MQIPT”](#) a pagina 1036.

Certificati e tecnologie di gestione dei certificati sono disponibili presso diversi fornitori terzi.

## Portachiavi di esempio MQIPT

I seguenti file keyring PKCS #12 di esempio vengono forniti con MQIPT nella directory secondaria `samples/ssl` per comodità durante la verifica:

### **sslSample.pfx**

Un keyring che contiene un certificato autofirmato di esempio.

### **sslCASample.pfx**

Un keyring del certificato CA di esempio.

Per accedere a questi file keyring di esempio, utilizzare la password `mqiptSample`.

Il certificato autofirmato di esempio deve essere utilizzato solo in un ambiente di test poiché le chiavi private del certificato è disponibile per tutti gli utenti MQIPT.

## Messaggi di errore SSL/TLS in MQIPT

Gli errori handshake vengono registrati nel log di connessione MQIPT sotto forma di eccezioni JSSE.

Per ulteriori informazioni, consultare [“Log di connessione in MQIPT”](#) a pagina 1059. La seguente tabella descrive le diverse eccezioni, la causa probabile e l'azione corrispondente per risolvere l'errore.

Le eccezioni di certificato di solito si riferiscono ai certificati all'estremità remota della connessione.

Se l'errore è correlato al certificato di un client IBM MQ o di un gestore code, il termine *file di chiavi* include il repository di chiavi IBM MQ del partner remoto.

In MQIPT, i certificati CA vengono memorizzati nel file key ring CA, identificato dalle proprietà di instradamento **SSLClientCAKeyRing** e **SSLServerCAKeyRing**. Se le proprietà di instradamento del key ring CA non sono impostate, i certificati CA vengono ricercati nel file keyring personale corrispondente (a cui fa riferimento la proprietà **SSLClientKeyRing** o **SSLServerKeyRing**).

Eccezione	Causa	Azione
CertificateException	Il certificato non è attendibile perché è firmato da una CA che non è nel keyring CA.	Controllare che tutti i certificati CA necessari siano presenti nel file keyring CA. Utilizzare lo strumento IBM Key Management fornito con MQIPT per aggiungere eventuali certificati CA mancanti, avendo cura di ottenere una copia di ciascun certificato CA da un'origine affidabile.
Eccezione CertificateExpired	<ol style="list-style-type: none"><li>1. Il certificato è scaduto: la sua data <b>notAfter</b> è passata.</li><li>2. L'orologio di sistema non è impostato correttamente.</li></ol>	<ol style="list-style-type: none"><li>1. Ottenere un nuovo certificato e inserirlo nel file keyring. Se il certificato appartiene a una CA (Certificate Authority), inserire il nuovo certificato nel file keyring CA.</li><li>2. Verificare che l'orologio di sistema UTC sia impostato sull'orario corretto.</li></ol>





Eccezione	Causa	Azione
Eccezione CertificateNotYetValid	<ol style="list-style-type: none"> <li>1. Il certificato viene utilizzato prematuramente: la relativa data <b>notBefore</b> non è ancora arrivata.</li> <li>2. L'orologio di sistema non è impostato correttamente.</li> </ol>	<ol style="list-style-type: none"> <li>1. Verificare che il certificato sia stato generato e firmato correttamente. Se la propria organizzazione gestisce la propria CA, l'orologio di sistema UTC per la CA potrebbe non essere corretto.</li> <li>2. Verificare che l'orologio di sistema UTC sia impostato sull'orario corretto.</li> </ol>
Eccezione CertificateParsing	<ol style="list-style-type: none"> <li>1. Il certificato contiene dati DER non validi.</li> <li>2. Il certificato utilizza funzioni DER non supportate.</li> </ol>	Assicurarsi che il certificato sia stato generato correttamente e che possa essere visualizzato nello strumento IBM Key Management fornito con MQIPT. Si consiglia di ottenere un nuovo certificato con un numero inferiore di estensioni certificato.
Eccezione CertificateRevoked	Il controllo della revoca del certificato è abilitato e il certificato è stato revocato.	Il certificato in questione non deve essere attendibile. Ottenere un certificato di sostituzione e verificare che il nuovo certificato e la chiave privata siano presenti nel file keyring.
CertPathBuilderException	La catena di certificati non è stata firmata da un'autorità di certificazione riconosciuta.	<ol style="list-style-type: none"> <li>1. Se si utilizzano certificati firmati dalla CA, verificare che tutti i certificati CA root e i certificati CA intermedi siano presenti nel file di chiavi della CA.</li> <li>2. Se si utilizzano certificati autofirmati, assicurarsi di aver estratto una copia della parte pubblica del certificato remoto e di aver aggiunto tale copia al file di chiavi CA. Evitare di utilizzare certificati autofirmati in ambienti di produzione.</li> </ol>

Eccezione	Causa	Azione
<p>Eccezione CertStore Eccezione KeyStore</p>	<p>Si è verificato un errore durante la lettura di un certificato da un keyring per uno dei seguenti motivi:</p> <ol style="list-style-type: none"> <li>1. Il file keyring è danneggiato.</li> <li>2. Manca il file key ring.</li> <li>3. La password memorizzata non corrisponde alla password del file key ring.</li> <li>4. Se l'instradamento è configurato per utilizzare l'hardware crittografico, MQIPT non è stato in grado di connettersi all'hardware crittografico.</li> </ol>	<ol style="list-style-type: none"> <li>1. Verificare che sia possibile leggere il file di chiavi e che tutti i certificati possano essere visualizzati con IBM Key Management Tool.</li> <li>2. Verificare che tutte le proprietà di instradamento key ring facciano riferimento al nome file corretto.</li> <li>3. Verificare che la password del file key ring memorizzata sia corretta. Utilizzare lo strumento <b>mqiptPW</b> per memorizzare la password corretta.</li> <li>4. Se l'instradamento è configurato per utilizzare l'hardware crittografico, controllare quanto segue: <ul style="list-style-type: none"> <li>• Il file delle proprietà di sicurezza Java specifica che il provider di sicurezza IBMPKCS11Impl è installato.</li> <li>• Il file delle proprietà di sicurezza Java contiene il nome completo del file di configurazione utilizzato per inizializzare il provider di sicurezza IBMPKCS11Impl .</li> <li>• Il file di configurazione utilizzato per inizializzare il provider di sicurezza IBMPKCS11Impl è valido.</li> </ul> </li> </ol>
<p>SSLException: Nessun certificato o chiave disponibile corrisponde alle suite di cifratura SSL abilitate.</p>	<p>È necessario disporre di un certificato personale con il tipo di chiave corretto per CipherSuites che si sta utilizzando. Ad esempio, CipherSuites i cui nomi iniziano con SSL_ECDH_ECDSA_ richiedono un certificato con una chiave pubblica Elliptic Curve. Le CipherSuites più comunemente utilizzate richiedono un certificato con una chiave pubblica RSA.</p>	<p>Aprire il file di chiavi con IBM Key Management Tool. Nella vista Certificati personali, selezionare ogni certificato e visualizzarlo. Fare clic su <b>Visualizza dettagli</b> e passare alla sezione Chiave pubblica oggetto per visualizzare il tipo di chiave pubblica. Quindi, controllare le proprietà di instradamento MQIPT <b>SSLClientCipherSuites</b> e <b>SSLServerCipherSuites</b> per assicurarsi che le CipherSuites appropriate siano abilitate.</p>

Eccezione	Causa	Azione
SSLException: nessuna suite di cifratura in comune SSLHandshakeException: nessuna suite di cifratura in comune	<p>L'handshake non è riuscito a concordare una CipherSuite perché non vi è alcuna sovrapposizione tra le serie di CipherSuites abilitate ad entrambi i lati della connessione. In particolare, una connessione IBM MQ in uscita abilita solo una singola cifratura in modo che gli instradamenti SSLServer MQIPT abbiano particolarmente probabilità di riscontrare questo errore.</p> <p>Questo errore può verificarsi anche quando si verificano tutte e tre le seguenti condizioni:</p> <ul style="list-style-type: none"> <li>• non è stata specificata alcuna CipherSuite sull'instradamento</li> <li>• non è possibile trovare alcun certificato del sito adatto nel keyring configurato per l'instradamento</li> <li>• I CipherSuites anonimi sono disabilitati</li> </ul>	<p>Controllare l'elenco di CipherSuites abilitati nelle proprietà di instradamento MQIPT <b>SSLClientCipherSuites</b> e <b>SSLServerCipherSuites</b> . Considerare l'abilitazione di ulteriori CipherSuites. Consultare la tabella fornita per determinare la CipherSuites corretta da abilitare per ciascun valore CipherSpec del canale IBM MQ .</p> <p>Se non viene specificato alcun CipherSuite sull'instradamento, controllare che le proprietà dell'instradamento del key ring facciano riferimento al file key ring corretto e che il key ring contenga un certificato personale che MQIPT può utilizzare. Se l'instradamento è configurato per utilizzare l'hardware di crittografia, verificare che l'attributo <b>tokenlabel</b> nel file di configurazione utilizzato per inizializzare il provider di sicurezza IBMPKCS11Impl specifici l'etichetta del token del dispositivo di crittografia corretta.</p>



## Gestione dei keystore MQIPT

  Utilizzare il comando **mqiptKeytool** per gestire certificati nei keystore utilizzati da IBM MQ Internet Pass-Thru (MQIPT).

Da IBM MQ 9.4.0, questo comando sostituisce il comando **mqiptKeycmd** utilizzato per gestire i certificati nelle versioni precedenti di MQIPT.

## Formato keystore richiesto per MQIPT

  MQIPT supporta i keystore che utilizzano il formato file PKCS #12 . Quando si utilizza il comando **mqiptKeytool** per gestire il keystore MQIPT , specificare il parametro **-storetype pkcs12** per indicare che il keystore utilizza il formato PKCS #12 .

MQIPT può inoltre accedere ai certificati memorizzati nell'hardware di crittografia che supporta l'interfaccia PKCS #11 . L'interfaccia può essere utilizzata anche per gestire certificati sull'hardware PKCS #11 . Per ulteriori informazioni, consultare [“Utilizzo dell'hardware di crittografia PKCS #11 in MQIPT” a pagina 1045.](#)

## Codifica della password del keystore per MQIPT

Crittografare la password del keystore in un formato che MQIPT può utilizzare per accedere al file. Per ulteriori informazioni, consultare [“Codifica di una password key ring in MQIPT” a pagina 1030.](#)

La funzione file stash supportata da IBM MQ non è supportata da MQIPT. Utilizzare il comando **mqiptPW** per codificare la password del keystore invece di utilizzare un file stash.

## Esempi



I seguenti esempi mostrano come il comando **mqiptKeytool** viene utilizzato per gestire i certificati in un keystore MQIPT .

- Il seguente comando crea un certificato personale autofirmato per scopi di test:

```
mqiptKeytool -genkeypair -keystore key.p12 -storetype pkcs12 -storepass password
 -alias mqipt -dname "CN=Test Certificate,OU=Sales,O=Example,C=US"
 -keyalg RSA -keysize 2048 -sigalg SHA256withRSA
```

Il comando crea un certificato digitale con una chiave pubblica RSA a 2048 bit e una firma digitale che utilizza RSA con algoritmo hash SHA-256 . Il certificato e le relative chiavi pubbliche e private associate sono memorizzati in un keystore in formato PKCS #12 denominato key . p12. Il file keystore viene creato se non esiste.

Quando si crea un certificato, scegliere un algoritmo di codifica della chiave pubblica, la dimensione della chiave e l'algoritmo di firma digitale che siano appropriati per le necessità di sicurezza della propria organizzazione. Per ulteriori informazioni, consultare [“Considerazioni sul certificato digitale per MQIPT”](#) a pagina 1036.

Questo esempio utilizza un certificato autofirmato adatto per scopi di test. In un ambiente di produzione, utilizzare invece un certificato firmato da un'autorità di certificazione.

- Il seguente comando crea una richiesta di certificato per un certificato firmato CA da utilizzare per scopi di produzione:

```
mqiptKeytool -certreq -keystore key.p12 -storetype pkcs12 -storepass password
 -alias mqipt -file cert.req
```

Il comando crea una CSR (Certificate Signing Request) in formato PKCS #10 . La CSR può essere inviata a un'autorità di certificazione per richiedere un certificato firmato CA. La coppia di chiavi pubblica e privata con l'alias mqipt deve essere creata prima di emettere questo comando, immettendo il comando **mqiptKeytool -genkeypair** nel precedente esempio.

- Il seguente comando riceve il file di certificato personale firmato CA denominato cert . crt nel keystore:

```
mqiptKeytool -importcert -keystore key.p12 -storetype pkcs12 -storepass password -file
cert.crt
```

Importare il certificato CA della CA che ha firmato il certificato personale nel keystore immettendo il seguente comando:

```
mqiptKeytool -importcert -keystore key.p12 -storetype pkcs12 -storepass password
 -file ca.crt -alias rootCA
```

### ***Considerazioni sul certificato digitale per MQIPT***

I punti da considerare includono la dimensione della chiave del certificato, la selezione di un algoritmo di firma digitale appropriato per il certificato e il certificato digitale e la compatibilità CipherSuite .

### **Considerazioni sulla dimensione della chiave del certificato per MQIPT**

La dimensione della chiave pubblica dipende dalla politica di sicurezza della propria organizzazione e dall'algoritmo di crittografia utilizzato. In generale, le dimensioni delle chiavi più grandi sono più sicure. La seguente tabella elenca le dimensioni chiave minime da utilizzare:

Algoritmo	Dimensione chiave minima (bit)
Curva ellittica	256
RSA	2048

**V 9.4.0** **V 9.4.0** Specifica la dimensione chiave del certificato utilizzando il parametro **-keysize** quando crei un certificato con il comando **mqiptykeytool**.

## Selezione di un algoritmo di firma digitale del certificato appropriato

Per prevenire la falsificazione di certificati digitali, è importante utilizzare un forte algoritmo di firma digitale. Quando si crea o si richiede un certificato, fare attenzione a selezionare un buon algoritmo.

Si consiglia di evitare l'utilizzo di vecchi algoritmi di firma digitale basati su MD5 o SHA-1 poiché tali algoritmi non sono considerati sicuri. Se possibile, utilizzare uno dei più recenti algoritmi di firma digitale basati su SHA-2 come SHA-256 con RSA (SHA256WithRSA).

**V 9.4.0** **V 9.4.0** Specificare l'algoritmo di firma utilizzando il parametro **-sigalg** quando viene creato un certificato con il comando **mqiptykeytool**.

## Certificato digitale e compatibilità CipherSuite in MQIPT

Non tutte le CipherSuites possono essere utilizzate con tutti i certificati digitali. Esistono vari tipi di CipherSuite, raggruppati in base al prefisso del nome CipherSuite. Ogni tipo di CipherSuite impone restrizioni diverse sul tipo di certificato digitale che può essere utilizzato. Queste limitazioni si applicano a tutte le connessioni MQIPT SSL/TLS, ma sono particolarmente rilevanti per gli utenti della crittografia Elliptic Curve. Durante un handshake socket sicuro, MQIPT seleziona automaticamente un certificato personale per identificare se stesso appropriato per la CipherSuite negoziata. Nella maggior parte dei casi, MQIPT interagisce automaticamente con il peer remoto. Tuttavia, in alcuni scenari potrebbe essere necessario utilizzare una MQIPT CipherSuite specifica per interagire con un sistema IBM MQ remoto.

**V 9.4.0** **V 9.4.0** Il comando **mqiptykeytool** può essere utilizzato per creare certificati con chiavi pubbliche DSA, RSA e Elliptic Curve. Consultare l'autorità di certificazione per informazioni sulla creazione di altri tipi di certificati.

Il tipo di certificato digitale da utilizzare dipende dal tipo di CipherSuite utilizzato:

- CipherSuites con nomi che iniziano con `SSL_ECDH_ECDSA_` e `SSL_ECDHE_ECDSA_` richiedono un certificato digitale con una chiave pubblica della curva ellittica.
- CipherSuites con nomi che contengono `anon` sono anonimi; non richiedono un certificato digitale per identificare il peer remoto. Tali CipherSuites possono evitare i sovraccarichi di gestione del ciclo di vita dei certificati nelle reti in cui viene utilizzato un mezzo alternativo di autenticazione, ma in generale, evitarne l'utilizzo a causa della mancanza di autenticazione.
- Altre CipherSuites richiedono un certificato digitale con chiave pubblica RSA.

## Uscita certificato in MQIPT

Lo scopo di un'uscita certificato è convalidare un certificato peer SSL/TLS ricevuto da MQIPT.

Puoi configurare un instradamento MQIPT per agire come un client SSL/TLS quando effettua una nuova connessione e per agire come un server SSL/TLS quando riceve una richiesta di connessione. Durante il processo di handshake SSL/TLS, un client SSL/TLS riceve un certificato peer dal server e il certificato può essere utilizzato per autenticare il server. Un server SSL/TLS può anche ricevere un certificato peer dal client e il certificato può essere utilizzato per autenticare il client.

L'uscita del certificato viene richiamata quando MQIPT riceve un certificato peer, consentendo di eseguire un'ulteriore convalida. Tutte le eccezioni rilevate dall'uscita vengono rilevate da MQIPT e la richiesta di connessione viene terminata. Si consiglia, quindi, all'uscita di rilevare tutte le eccezioni e di restituire un codice di ritorno appropriato a MQIPT.

Viene fornito un esempio per mostrare che un'uscita certificato può essere implementata per ulteriori informazioni consultare [Utilizzo di un'uscita certificato per autenticare un server SSL/TLS](#).

**Nota:** MQIPT viene eseguito in un singolo Java Virtual Machine in modo che un'uscita di certificato definita dall'utente possa compromettere il normale funzionamento di MQIPT in uno dei seguenti modi:

- Influisce sulle risorse di sistema
- Genera colli di bottiglia
- Prestazioni ridotte

È necessario verificare ampiamente gli effetti dell'uscita del certificato prima di implementarlo in un ambiente di produzione.

### ***La classe com.ibm.mq.ipt.exit.CertificateExit in MQIPT***

La classe `com.ibm.mq.ipt.exit.CertificateExit` è una classe astratta che deve essere implementata dalla classe definita con la proprietà `SSLExitName`.

La classe contiene le implementazioni predefinite per l'esecuzione dell'uscita e alcuni metodi pubblici che è possibile sovrascrivere, in base ai propri requisiti. L'elenco completo dei metodi supportati è il seguente:

## **Metodi**

### **init int pubblico (IPTTrace)**

Il metodo `init` viene richiamato da MQIPT quando l'uscita viene caricata da MQIPT e può essere implementato per eseguire qualsiasi inizializzazione dell'uscita; ad esempio, il caricamento dei dati utilizzati durante il processo di convalida. L'implementazione predefinita non fa nulla.

### **aggiornamento int pubblico (IPTTrace)**

Il metodo di aggiornamento è implementato per eseguire un aggiornamento di qualsiasi dato; ad esempio, il ricaricamento di qualsiasi dato per il disco utilizzato durante il processo di convalida. Questo metodo viene richiamato quando l'amministratore MQIPT ha immesso un comando di aggiornamento. L'implementazione predefinita non fa nulla.

### **public void close (IPTTrace)**

Il metodo di chiusura viene implementato per eseguire le operazioni di manutenzione quando l'instradamento sta per essere arrestato o MQIPT sta per essere chiuso. L'implementazione predefinita non fa nulla.

### **Convalida risposta CertificateExitpubblica (IPTTrace)**

Il metodo di convalida viene richiamato per eseguire la convalida del certificato peer. L'oggetto di ritorno può essere utilizzato per passare le informazioni a MQIPT; ad esempio, un codice di ritorno e del testo che può essere aggiunto al log di connessione. L'implementazione predefinita restituisce una risposta `CertificateExit` con `CertificateExitResponse.OK`.

Metodi supportati per ottenere le proprietà:

### **public int getListenerPorta ()**

richiama la porta del listener di instradamento - come definito dalla proprietà `ListenerPort`

### **public String getDestination()**

richiama l'indirizzo di destinazione - come definito dalla proprietà `Destinazione`

### **public int getDestinationPort ()**

richiama l'indirizzo della porta del listener di destinazione - come definito dalla proprietà `DestinationPort`

### **public String getClientIPAddress ()**

richiama l'indirizzo IP del client che effettua la richiesta di collegamento

### **public int getClientPortAddress()**

richiama l'indirizzo porta utilizzato dal client che effettua la richiesta di connessione

**booleano pubblico isSSLClient()**

utilizzato per determinare se l'uscita viene richiamata come un client SSL/TLS o un server SSL/TLS. Se questo valore restituisce true, l'uscita si trova sul lato client della connessione, convalidando il certificato ottenuto dal server. Se questo restituisce false, l'uscita si trova sul lato server della connessione, convalidando il certificato inviato dal client. È valido per un instradamento che funga sia da server SSL/TLS che da un client SSL/TLS, decodificando e ricodificando il traffico. In questa situazione, sebbene esista una singola classe di uscita, alcune istanze della classe verranno richiamate come client e alcune come server. È possibile utilizzare isSSLClient per determinare la situazione per una determinata istanza.

**public int getConnThreadID()**

utilizzato per richiamare l'ID del thread di lavoro che sta gestendo la richiesta di connessione, che può essere utile per il debug.

**public String getChannelNome ()**

richiama il nome del canale di IBM MQ utilizzato nella richiesta di connessione. Questa opzione è disponibile solo quando la richiesta in entrata non utilizza SSL/TLS e MQIPT funge da client SSL/TLS.

**public String getQMName()**

richiama il nome del gestore code IBM MQ utilizzato nella richiesta di connessione. Questa opzione è disponibile solo quando la richiesta client non utilizza SSL/TLS e MQIPT funge da client SSL/TLS.

**Booleano pubblico getTimeout()**

utilizzato dall'uscita per determinare se il timeout è scaduto.

**public IPTCertificate getCertificate()**

richiama il certificato SSL/TLS che deve essere convalidato.

**public String getExitData ()**

richiama i dati di uscita, come definito dalla proprietà SLExitData .

**public String getExitName ()**

richiama il nome uscita, come definito dalla proprietà SLExitName .

***La classe com.ibm.mq.ipt.exit.CertificateExitResponse in MQIPT***

Questa classe viene utilizzata per passare le informazioni a MQIPT dopo che un certificato è stato convalidato.

**Costruttori****public CertificateExitResponse (int rc, messaggio stringa)**

Questo costruttore può essere utilizzato per restituire un codice di ritorno e del testo del messaggio. I codici di errore possibili sono

- ExitRc.OK
- ExitRc.ERRORRE\_CONVALIDA
- ExitRc.CONVALIDA\_RIFIUTATA

**Risposta CertificateExitpubblica (int rc)**

Questo costruttore può essere utilizzato per restituire un codice di ritorno, senza testo del messaggio. I codici di errore possibili sono

- ExitRc.OK
- ExitRc.ERRORRE\_CONVALIDA
- ExitRc.CONVALIDA\_RIFIUTATA

**Risposta () CertificateExitpubblica**

Questo costruttore può essere utilizzato per passare il codice di ritorno ExitRc.OK, senza testo del messaggio.

## Metodi

### **public String getVersion()**

Questo metodo restituisce la versione di questa classe.

### **public String toString**

Questo metodo restituirà una rappresentazione di stringa della risposta, ad esempio, " Codice di errore: 4, Messaggio: Controllo CRL non riuscito.

## ***La classe com.ibm.mq.ipt.exit.IPTCertificate in MQIPT***

Questa classe contiene il certificato SSL/TLS da convalidare.

## Metodi

### **public int getVersion()**

Questo metodo restituisce la versione di questa classe.

### **public byte [] getDerCodifica ()**

Questo metodo restituisce la codifica ASN.1/DER del certificato X.509 o NULL se si verifica un errore.

### **byte pubblico [] getPemCodifica ()**

Questo metodo restituisce la codifica PEM (BASE64) del certificato X.509 o NULL se si verifica un errore.

### **public String getLabel()**

Questo metodo restituisce l'etichetta del certificato o NULL se si verifica un errore.

### **public String getName()**

Questo metodo restituisce il DN (Distinguished Name) del certificato o NULL se non disponibile. Ad esempio:

```
CN=Test Queue Manager,OU=Sales,O=Example,L=London,C=GB
```

### **public String getIssuerNome ()**

Questo metodo restituisce il DN (Distinguished Name) dell'emittente del certificato o NULL se non disponibile. Ad esempio:

```
CN=Certificate Authority,OU=Security,O=Example,L=New York,C=US
```

### **pubblico IPTCertificate getSigner()**

Questo metodo restituisce il certificato del firmatario o NULL se non è disponibile. Per un certificato autofirmato, restituirà un riferimento a se stesso.

### **public String toString()**

Questo metodo restituisce una rappresentazione stringa del certificato.

## ***La classe com.ibm.mq.ipt.exit.IPTTrace in MQIPT***

Le funzioni di traccia MQIPT forniscono chiamate di entrata e di uscita, che possono essere utilizzate all'entrata e all'uscita da un metodo. Ci sono anche varie chiamate di dati per tracciare informazioni utili.

## Metodi

### **public void entry (String fid)**

Dove *fid* viene utilizzato per identificare dove è stata effettuata la chiamata, ad esempio il nome della classe e del metodo.



Questo metodo scrive una voce nel file di output di traccia con il livello di rientro appropriato per registrare il punto in cui il flusso di controllo entra in un metodo. Questa chiamata è facoltativa, ma se viene utilizzata, è necessario utilizzare anche una chiamata corrispondente a "exit (String)" all'interno dello stesso metodo.

#### **public void exit (String *fid*)**

Dove *fid* viene utilizzato per identificare dove è stata effettuata la chiamata, ad esempio il nome della classe e del metodo.

Questo metodo scrive un'uscita nel file di output di traccia con il livello di rientro appropriato per registrare il punto in cui il flusso di controllo lascia un metodo. Questo metodo viene utilizzato solo quando una chiamata a "entry (String)" è stata precedentemente utilizzata all'interno dello stesso metodo.

#### **public void exit (String *fid*, int *rc*)**

Dove *fid* viene utilizzato per identificare il punto in cui è stata effettuata la chiamata, ad esempio il nome della classe e del metodo e *rc* è il codice di ritorno numerico dal metodo. Questo metodo di tracciamento deve essere utilizzato per registrare l'uscita dai metodi che restituiscono un numero intero.

Questo metodo scrive un'uscita nel file di output di traccia con il livello di rientro appropriato per registrare il punto in cui il flusso di controllo lascia un metodo e il codice di ritorno numerico da tale metodo. Questo metodo viene utilizzato solo quando una chiamata a "entry (String)" è stata precedentemente utilizzata all'interno dello stesso metodo.

#### **public void exit (String *fid*, boolean *rc*)**

Dove *fid* viene utilizzato per identificare dove è stata effettuata la chiamata, ad esempio il nome della classe e del metodo e *rc* è il codice di ritorno booleano dal metodo. Questo metodo di traccia deve essere utilizzato per registrare l'uscita dai metodi che restituiscono un valore booleano.

Questo metodo scrive un'uscita nel file di output di traccia con il livello di rientro appropriato per registrare il punto in cui il flusso di controllo lascia un metodo e il codice di ritorno booleano da tale metodo. Questo metodo viene utilizzato solo quando una chiamata a "entry (String)" è stata precedentemente utilizzata all'interno dello stesso metodo.

#### **public void data (String *fid*, String *data*)**

Dove *fid* viene utilizzato per identificare dove è stata effettuata la chiamata, ad esempio il nome della classe e del metodo.

Questo metodo scrive alcuni dati stringa nel file di output di traccia.

#### **public void data (String *fid*, int *data*)**

Dove *fid* viene utilizzato per identificare dove è stata effettuata la chiamata, ad esempio il nome della classe e del metodo.

Questo metodo scrive alcuni dati interi nel file di output di traccia.

#### **public void data (String *fid*, byte [])**

Dove *fid* viene utilizzato per identificare dove è stata effettuata la chiamata, ad esempio il nome della classe e del metodo.

Questo metodo scrive alcuni dati binari nel file di output di traccia.

### **Traccia di esempio**

Per facilitare la diagnosi dei problemi in un'uscita, è possibile utilizzare la stessa funzione di traccia di MQIPT, oppure è possibile implementare le proprie funzioni di traccia. Se si decide di utilizzare le funzioni di traccia MQIPT, sono presenti chiamate di entrata e di uscita, che possono essere utilizzate all'entrata

e all'uscita da un metodo. Esistono anche diverse chiamate di dati per tracciare informazioni utili come mostrato nel seguente esempio.

```

/**
 * This method is called to initialize the exit (for example, for
 * loading validation information) and place itself in a ready
 * state to validate connection requests.
 */
public int init(IPTTrace t) {
 final String fid = "MyExit.init";

 // Trace entry into this method
 t.entry(fid);

 // Trace useful information
 t.data(fid, "Starting exit - MQIPT version " + getVersion());

 // Perform initialization and load any data
 t.data(fid, "Ready for work");

 // Trace exit from this method
 t.exit(fid);

 return ExitRc.OK;
}

```

Questo metodo produce la traccia nel formato mostrato nel seguente esempio:

```

16:36:48.625 14 5000-1s -----{ ConnectionThread.setCertificateExit()
16:36:48.625 14 5000-1s Creating instance of certificate exit
16:36:48.625 14 5000-1s Calling init() of certificate exit
16:36:48.625 14 5000-1s -----} MyExit.init()
16:36:48.625 14 5000-1s Starting exit - MQIPT version 2.1.0.0
16:36:48.625 14 5000-1s Ready for work
16:36:48.625 14 5000-1s -----} MyExit.init() rc=0
16:36:48.625 14 5000-1s -----} ConnectionThread.setCertificateExit() rc=0

```

### **Codici di ritorno di uscita certificato in MQIPT**

I codici di ritorno che MQIPT riconosce quando richiama un'uscita certificato in un numero di situazioni differenti

I seguenti codici di ritorno vengono riconosciuti da MQIPT quando si richiama un'uscita certificato nelle situazioni seguenti:

<b>Codice di ritorno</b>	<b>Descrizione</b>	<b>inizializ zare</b>	<b>Convali da</b>	<b>Aggiorn a</b>
ExitRc.OK	Richiesta completata correttamente.	sì	sì	sì
ExitRc.ERRORE DI INIZIALIZZAZIONE	Richiesta di inizializzazione non riuscita, l'instradamento verrà disabilitato.	sì		
ExitRc.ERRORE_RISPEDIZIONE	Richiesta di aggiornamento non riuscita, l'instradamento verrà disabilitato.			sì
ExitRc.ERRORE_CONVALIDA	Processo di convalida non riuscito, richiesta di connessione rifiutata.		sì	
ExitRc.CONVALIDA_RIFIUTATA	Richiesta di convalida rifiutata, richiesta di connessione rifiutata.		sì	

### **LDAP e CRL in MQIPT**

MQIPT supporta l'utilizzo di un server LDAP (Lightweight Directory Access Protocol) per eseguire l'autenticazione CRL (Certificate Revocation List) su un certificato digitale.

Il supporto LDAP è stato implementato in modo simile a quello in IBM MQ, poiché lo stesso server LDAP può essere utilizzato sia per IBM MQ che per MQIPT.

Durante l'handshake SSL/TLS, i partner comunicanti si autenticano reciprocamente con i certificati digitali. L'autenticazione può includere una conferma che il certificato ricevuto sia ancora sicuro. Le autorità di certificazione (CA) revocano i certificati per vari motivi, tra cui:

- Il proprietario è stato spostato in un'organizzazione diversa.
- La chiave privata non è più segreta.

Le CA pubblicano i certificati personali revocati in un CRL (Certificate Revocation List). I certificati AC revocati vengono pubblicati in un elenco ARL (Authority Revocation List). Notare che i riferimenti successivi ai CRL si applicano anche agli ARL.

Per ulteriori informazioni sull'utilizzo dei server LDAP con IBM MQ e sulla gestione di CRL e ARL, consultare [Gestione degli elenchi di revoca dei certificati e degli elenchi di revoca delle autorizzazioni](#).

MQIPT può supportare fino a due server LDAP su ciascun instradamento. Il primo server LDAP viene considerato come il server principale con il secondo server LDAP mantenuto come backup. Il secondo server viene utilizzato solo se non è possibile raggiungere il server principale. Il server di backup deve essere un'immagine di mirroring del server principale.

L'accesso alle informazioni memorizzate su un server LDAP può essere protetto con un ID utente e una password utilizzando le proprietà ID utente e password LDAP. Le password del server LDAP possono essere codificate nella configurazione MQIPT da IBM MQ 9.1.5. Per ulteriori informazioni sulla codifica delle password che devono essere utilizzate da MQIPT, consultare [“Crittografia delle password memorizzate in MQIPT”](#) a pagina 1056.

Quando MQIPT carica un token PKCS #12 da un file key ring, viene verificata la validità CRL di qualsiasi certificato CA. Se il certificato CA ha un CRL allegato, viene controllato per verificare se è scaduto e, in tal caso, viene richiamato un CRL più recente dal server LDAP. Qualsiasi CRL richiamato viene caricato nel token corrente e collegato al relativo certificato CA.

Se non ci sono voci che corrispondono alla CA fornita quando una query viene inviata al server LDAP principale, si presuppone che non ci siano CRL per quella CA e che il server di backup non venga utilizzato. Tuttavia, se il server LDAP principale non può essere raggiunto o non viene restituito entro un determinato intervallo di tempo, viene utilizzato il server di backup. Eventuali errori del server di backup causano la chiusura della connessione client. Questa azione può essere sovrascritta impostando la proprietà **LDAPIgnoreErrors** su true.

I CRL richiamati da MQIPT vengono conservati in una cache e condivisi da tutte le connessioni su tale instradamento. Se un CRL memorizzato nella cache è scaduto, il CRL viene rimosso dalla cache e ne viene richiamato uno nuovo dal server LDAP. Se un nuovo CRL non è disponibile, la connessione viene ancora rifiutata.

Viene controllata anche la scadenza di un CRL richiamato dal server LDAP e viene visualizzato un messaggio di avviso (MQCPW001). Il CRL scaduto è ancora caricato nel sistema e tutte le richieste di connessione che fanno riferimento a quel CRL vengono rifiutate. Sostituire il CRL scaduto nel server LDAP con uno corrente.

La proprietà **LDAPCacheTimeout** può essere utilizzata per controllare la frequenza con cui la cache CRL viene cancellata. Il valore predefinito è 1 giorno. Se si imposta questo valore su 0, le voci della cache non vengono cancellate fino a quando l'instradamento non viene riavviato.

Un CRL scaduto può essere memorizzato in un file key ring o su un server LDAP. Se non è stato emesso un nuovo CRL, vengono rifiutate ulteriori richieste di connessione. È possibile ignorare i CRL scaduti abilitando la proprietà **IgnoreExpiredCRLs**.

**Nota:** Se si abilita la proprietà **LDAPIgnoreErrors** o la proprietà **IgnoreExpiredCRLs**, è possibile utilizzare un certificato revocato per stabilire una connessione SSL/TLS.

## Proprietà OU del certificato con più valori in MQIPT

È possibile mettere in corrispondenza più valori OU (organizational unit) nel certificato DN (Distinguished Name).

Le seguenti proprietà di instradamento ora supportano la corrispondenza di più valori OU:

- **SSLClientDN\_OU**
- **SSLClientSiteDN\_OU**
- **SSLServerDN\_OU**
- **SSLServerSiteDN\_OU**

Per far corrispondere più valori OU, utilizzare una virgola come separatore nel valore della proprietà di instradamento. Ad esempio:

```
SSLClientDN_OU=Sales, Europe
```

Questo corrisponde ai certificati con OU=Sales e OU=Europe. I valori OU corrispondono nella stessa sequenza di più valori OU nei filtri IBM MQ SSLPEER.

Non specificare la stessa proprietà di instradamento più di una volta nella sezione [route] . Il modo corretto per far corrispondere più valori OU è specificare la proprietà una volta, come mostrato nell'esempio precedente. Se si immette lo stesso attributo più di una volta nella stessa sezione mqipt.conf, l'ultimo valore diventa effettivo. Ad esempio, le seguenti voci risulterebbero solo nella corrispondenza di Europa perché la seconda riga sovrascrive la prima:

```
SSLClientDN_OU=Sales
SSLClientDN_OU=Europe
```

Se è necessario mettere in corrispondenza una virgola letterale all'interno di un valore OU, inserire una barra retroversa (\) come carattere escape immediatamente prima della virgola. Ad esempio:

```
SSLClientDN_OU=Sales\, Europe
```

Corrisponde a un singolo valore: OU=Sales, Europe. Una barra rovesciata che non è immediatamente seguita da una virgola corrisponde a una barra rovesciata letterale.

Se si sta eseguendo l'aggiornamento da una release precedente di MQIPT e si basa sulla capacità di mettere in corrispondenza virgole nei valori OU, è necessario inserire caratteri di escape barra retroversa nelle proprietà di instradamento OU per preservare il comportamento precedente.

### **Deprecated** Abilitazione di protocolli obsoleti e suite di crittografia in MQIPT

Per impostazione predefinita, i protocolli dei socket protetti e le suite di cifratura considerate non sicure sono disabilitati nel Java runtime environment (JRE) fornito con MQIPT. Questi protocolli e suite di cifratura obsoleti devono essere abilitati prima di poter essere utilizzati.

## Informazioni su questa attività

Se si è consapevoli dei potenziali rischi, ma è ancora necessario utilizzare uno dei protocolli o delle suite di cifratura considerati non sicuri in MQIPT, seguire questa procedura per abilitare il protocollo o la suite di cifratura che è necessario utilizzare.

**Nota:** I protocolli e le suite di cifratura obsoleti non possono essere utilizzati con la porta comandi TLS.

## Procedura

1. Modificare il file `java.security`, che si trova nella directory `mqipt_path/java/jre/lib/security`, dove `mqipt_path` è il percorso in cui è installato MQIPT.
2. Aggiungere il supporto al JRE per un protocollo o un algoritmo rimuovendo la voce corrispondente dall'elenco di algoritmi disabilitati nella proprietà `jdk.tls.disabledAlgorithms`.
  - Per aggiungere il supporto per un protocollo, rimuovere il protocollo dall'elenco degli algoritmi disabilitati. Ad esempio, per aggiungere supporto per TLS 1.0, rimuovere TLSv1 dall'elenco.

- Per aggiungere supporto per una suite di cifratura, rimuovere gli algoritmi corrispondenti dall'elenco degli algoritmi disabilitati. Ad esempio, per aggiungere il supporto per la suite di crittografia `SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA`, rimuovere `3DES_EDE_CBC` e `DESede` dall'elenco.
3. Per abilitare SSL 3.0 in JRE, è necessario impostare anche le proprietà del sistema `com.ibm.jsse2.disableSSLv3=false`.

Se si sta avviando MQIPT dalla riga comandi utilizzando il comando `mqipt`, è possibile impostare la proprietà utilizzando la variabile di ambiente `MQIPT_JVM_OPTIONS`. Ad esempio:

```
set MQIPT_JVM_OPTIONS=-Dcom.ibm.jsse2.disableSSLv3=false
```

**Windows** Se MQIPT è installato come un servizio Windows, è possibile impostare la proprietà definendo un valore stringa nel registro Windows nella chiave `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MQInternetPassThru`. Il valore deve avere i seguenti attributi:

**Nome**

Opzioni MqiptJvm

**Dati valore**

`-Dcom.ibm.jsse2.disableSSLv3=false`

4. Per abilitare SSL 3.0, TLS 1.0 o TLS 1.1 su un instradamento MQIPT, aggiungi il protocollo corrispondente alla proprietà di instradamento `SSLServerProtocols` o `SSLClientProtocols`.
5. Riavviare MQIPT per rendere effettive le modifiche alle proprietà JRE.

## Utilizzo dell'hardware di crittografia PKCS #11 in MQIPT

MQIPT può accedere ai certificati digitali memorizzati nell'hardware crittografico che supporta l'interfaccia PKCS #11.

### Prima di iniziare

Prima di configurare MQIPT per l'utilizzo dell'hardware crittografico, assicurarsi che la scheda crittografica, il driver della scheda e qualsiasi software di supporto associato siano installati e funzionino correttamente.

Il supporto per l'hardware di crittografia PKCS #11 in MQIPT è fornito dal provider di crittografia IBM Java PKCS11 (provider `IBMPKCS11Impl`). Per ulteriori informazioni relative al provider `IBMPKCS11Impl` e all'elenco di schede crittografiche supportate da Java 8, consultare [IBM PKCS11 Cryptographic Provider](#).

### Informazioni su questa attività

È possibile memorizzare i certificati personali e i certificati CA che MQIPT utilizza in un keystore hardware crittografico. Tuttavia, poiché una periferica PKCS #11 normalmente non dispone di spazio sufficiente per memorizzare molti certificati del firmatario, è possibile utilizzare un keystore basato su file separato per i certificati CA.

Seguire questa procedura per configurare MQIPT per utilizzare i certificati in un keystore hardware crittografico.

**Nota:**

- L'utilizzo dell'hardware crittografico con MQIPT è una funzionalità IBM MQ Advanced. Per utilizzare questa funzionalità, è necessario che il gestore code locale connesso alla rotta MQIPT disponga anche della titolarità IBM MQ Advanced, IBM MQ Appliance, IBM MQ Advanced for z/OS VUE o IBM MQ Advanced for z/OS.
- **V9.4.0** Il comando `mqiptKeytool` non supporta attualmente l'hardware crittografico PKCS #11. Questa procedura utilizza il comando `ikeycmd` fornito con Java runtime environment (JRE) per gestire i certificati nell'hardware crittografico. Assicurarsi che il comando `ikeycmd` nel JRE nel percorso di installazione MQIPT sia utilizzato per eseguire questi comandi.

## Procedura

1. Creare il file di configurazione utilizzato quando viene inizializzato il provider IBMPKCS11Impl .

Scaricare i file di configurazione di esempio per ciascuna delle schede crittografiche hardware supportate dal provider IBMPKCS11Impl e configurare un esempio per il sistema. Gli esempi possono essere scaricati dal seguente argomento in IBM Documentation per Java: [File di configurazione](#).

Il file di configurazione è un file di testo e deve contenere almeno i seguenti attributi:

**name**

Il suffisso del nome dell'istanza provider.

**library**

Il nome completo della libreria PKCS #11 fornita con l'hardware crittografico.

**tokenlabel**

L'etichetta del token dell'unità crittografica PKCS #11 .

Ad esempio, il file di configurazione potrebbe contenere le seguenti voci:

```
name = IPTPKCS11Provider
library = /usr/lib64/pkcs11/PKCS11_API.so
tokenlabel = icatoken
```

2. Modificare il file delle proprietà di sicurezza Java , `java.security`, che si trova nella sottodirectory `java/jre/lib/security` della directory di installazione MQIPT .

- a) Se non è già presente nel file, aggiungere il provider di sicurezza IBMPKCS11Impl .

Ad esempio, aggiungendo la seguente riga:

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
```

- b) Aggiungere il nome completo del file di configurazione dopo il nome provider.

Ad esempio, se il file di configurazione creato nel passo "1" a pagina 1046 è denominato `/opt/mqipt/pkcs11.cfg`, aggiungere questo percorso alla stessa riga del provider di sicurezza:

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/mqipt/
pkcs11.cfg
```

3. Richiedere un certificato personale per l'hardware crittografico.

Immettere il seguente comando per creare una richiesta di certificato con il comando **ikeycmd** :

```
MQIPT_INSTALL_DIRECTORY/java/jre/bin/ikeycmd -certreq -create
-crypto module_name -tokenlabel hardware_token
-pw password -label label -size key_size
-sig_alg algorithm -dn distinguished_name -file filename
```

dove:

**DIRECTORY\_INSTALL\_MQIP**

È la directory di installazione di MQIPT .

**-crypto nome\_modulo**

Specifica il nome completo della libreria PKCS #11 fornita con l'hardware di crittografia.

**-tokenlabel etichetta\_token**

Specifica l'etichetta del token dell'unità crittografica PKCS #11 .

**-pw password**

Specifica la parola d'ordine per accedere all'hardware crittografico.

**-label label**

Specifica l'etichetta del certificato.

**-size dimensione\_chiave**

Specifica la dimensione della chiave. Il valore può essere 512, 1024, 2048o 4096.

**-sig\_alg algoritmo**

Specifica l'algoritmo di firma asimmetrico utilizzato per creare la coppia di chiavi della voce. Il valore può essere MD2\_WITH\_RSA, MD2WithRSA, MD5\_WITH\_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256\_WITH\_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384\_WITH\_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512\_WITH\_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA\_WITH\_DSA, SHA\_WITH\_RSAo SHAWithDSA. Il valore predefinito è SHA256WithRSA.



**-dn nome\_distinto**

Specifica il nome distinto X.500 racchiuso tra virgolette.

**-file nomefile**

Specifica il nome file per la richiesta di certificato.

4. Dopo che la CA ha inviato il certificato personale, aggiungere il certificato CA al keystore hardware di crittografia o a un file keystore CA, se non è già presente.

-   Per aggiungere il certificato CA ad un file keystore CA PKCS #12 , immettere il seguente comando:

```
mqiptKeytool -importcert -keystore filename -storetype pkcs12 -storepass password
 -alias label -file cert_filename
```

dove:

**-keystore nomefile**

Specifica il nome del file keystore CA. Il keystore viene creato se non esiste già.

**-storepass password**

Specifica la password del keystore CA.

**-alias etichetta**

Specifica l'etichetta del certificato CA.

**-file nomefile\_cert**

Specifica il nome del file che contiene il certificato CA.

- Per aggiungere il certificato CA all'hardware crittografico, immettere il seguente comando:

```
MQIPT_INSTALL_DIRECTORY/java/jre/bin/ikeycmd -cert -add
 -crypto module_name -tokenlabel hardware_token
 -pw password -label label -file cert_filename
```

dove:

**DIRECTORY\_INSTALL\_MQIP**

È la directory di installazione di MQIPT .

**-crypto nome\_modulo**

Specifica il nome completo della libreria PKCS #11 fornita con l'hardware di crittografia.

**-tokenlabel etichetta\_token**

Specifica l'etichetta del token dell'unità crittografica PKCS #11 .

**-pw password**

Specifica la parola d'ordine per accedere all'hardware crittografico.

**-label label**

Specifica l'etichetta del certificato CA.

**-file nomefile\_cert**

Specifica il nome del file che contiene il certificato CA.

5. Ricevere il certificato personale, fornito dalla CA, nel keystore hardware crittografico.

Immettere il seguente comando per aggiungere il certificato al keystore hardware crittografico:

```
MQIPT_INSTALL_DIRECTORY/java/jre/bin/ikeycmd -cert -receive
 -crypto module_name -tokenlabel hardware_token
 -pw password -file filename
```

dove:

**DIRECTORY\_INSTALL\_MQIP**

È la directory di installazione di MQIPT .

**-crypto nome\_modulo**

Specifica il nome completo della libreria PKCS #11 fornita con l'hardware di crittografia.

**-tokenlabel etichetta\_token**

Specifica l'etichetta del token dell'unità crittografica PKCS #11 .

**-pw password**

Specifica la parola d'ordine per accedere all'hardware crittografico.

**-label label**

Specifica l'etichetta del certificato.

**-file nomefile\_cert**

Specifica il nome del file che contiene il certificato da aggiungere.

Se il certificato CA è memorizzato in un file keystore CA, piuttosto che nell'hardware crittografico, il comando emette un'avvertenza che indica che la catena di certificati non può essere convalidata poiché non è possibile accedere al keystore CA quando il certificato personale viene aggiunto all'hardwarekeystore crittografico.

6. Codificare la parola d'ordine per accedere all'hardware crittografico 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 per accedere all'hardware crittografico da codificare quando richiesto.

Per ulteriori informazioni sulla codifica delle password del keystore, consultare [“Codifica di una password key ring in MQIPT”](#) a pagina 1030.

7. Se il certificato CA è stato aggiunto a un file keystore nel passo [“4”](#) a pagina 1047, seguire le istruzioni nel passo [“6”](#) a pagina 1048 per codificare la password per il keystore CA.
8. Modificare il file di configurazione `mqipt.conf` .
  - a) Confermare di disporre della titolarità appropriata per utilizzare questa funzione IBM MQ Advanced impostando la proprietà globale **EnableAdvancedCapabilities** su `true`.
  - b) Abilitare l'utilizzo del keystore dell'hardware crittografico sull'instradamento impostando una o più proprietà **SSLServerKeyRingUseCryptoHardware**, **SSLServerCAKeyRingUseCryptoHardware**, **SSLServerKeyRingUseCryptoHardware** o **SSLServerKeyRingUseCryptoHardware** su `true`.

Per ulteriori informazioni sulle proprietà per abilitare l'utilizzo dell'hardware crittografico su un instradamento, consultare [MQIPT route properties](#).

È anche possibile utilizzare l'hardware di crittografia con la porta comandi TLS impostando la proprietà **SSLCommandPortKeyRingUseCryptoHardware** su `true`.
  - c) Se si utilizza un file keystore per i certificati CA, specificare l'ubicazione del keystore CA impostando una o più proprietà **SSLServerCAKeyRing** o **SSLServerCAKeyRing** .

Se si configura l'instradamento per utilizzare l'hardware crittografico per il certificato del sito e non si specifica un file keyring CA, il keystore dell'hardware crittografico viene utilizzato come keystore CA.
  - d) Specificare la password crittografata per accedere all'hardware crittografico e al keystore CA utilizzando la proprietà **SSLServerKeyRingPW**, **SSLServerCAKeyRingPW**, **SSLClientKeyRingPW**, **SSLClientCAKeyRingPW** o **SSLCommandPortKeyRingPW** .

Impostare il valore delle proprietà **SSL\*KeyRingPW** sulla password crittografata creata dal comando **mqiptPW** .



e) Se l'hardware crittografico contiene più di un certificato personale, specificare quale certificato MQIPT invia al server o al client SSL/TLS per l'autenticazione.

- Per un instradamento client SSL/TLS, specificare quale certificato viene selezionato utilizzando una o più proprietà **SSLClientSite\***.
- Per un instradamento server SSL/TLS, specificare quale certificato viene selezionato utilizzando una o più proprietà **SSLServerSite\***.
- Per la porta comandi TLS, specificare il certificato selezionato utilizzando la proprietà **SSLCommandPortSiteLabel** per specificare l'etichetta del certificato.

Per ulteriori informazioni sulla selezione dei certificati da un keystore, consultare [“Selezione di certificati da un file di chiavi in MQIPT” a pagina 1030](#). Le proprietà per selezionare un certificato da un keystore sono descritte in [MQIPT route properties](#).

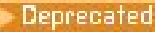
Ad esempio, per utilizzare un keystore hardware crittografico per il certificato del sito su un instradamento del server TLS e un file keystore per memorizzare i certificati CA per lo stesso instradamento, aggiungere le seguenti proprietà alla definizione di instradamento:

```
SSLServerKeyRingUseCryptoHardware=true
SSLServerKeyRingPW=<mqiptPW>1!g0RdM4wft5d1rCgNMDEGag==!dZxhgQD2A8Ea0yeqawQvPg==
SSLServerCAKeyRing=/opt/mqipt/ssl/ca.pfx
SSLServerCAKeyRingPW=<mqiptPW>1!3Vdrpiu6kMwn0sWRCVgT5g==!LH1tGLEg30FvN8+02Re0YA==
SSLServerSiteLabel=mqiptsite
```

9. Riavviare MQIPT.

## Java security manager in MQIPT

Java security manager può essere utilizzato con qualsiasi funzione MQIPT per fornire un ulteriore livello di sicurezza.

**Nota:**  L'utilizzo di Java security manager con MQIPT è obsoleto poiché Java security manager è obsoleto per la rimozione in una release futura di Java.

MQIPT utilizza il valore predefinito Java security manager come definito nella classe `java.lang.SecurityManager`. La funzione Java security manager in MQIPT può essere abilitata o disabilitata utilizzando la proprietà globale **SecurityManager**. Per ulteriori informazioni, consultare [MQIPT proprietà globali](#).

Java security manager utilizza due file di politica predefiniti:

- Un file della politica del sistema globale denominato `$MQIPT_PATH/java/jre/lib/security/java.policy` (dove `$MQIPT_PATH` è la directory in cui è installato MQIPT) viene utilizzato da tutte le istanze di una macchina virtuale su un host.
- Un file della politica specifico per l'utente denominato `.java.policy`, che può esistere nella directory home dell'utente.

È anche possibile utilizzare un ulteriore file della politica MQIPT. Si consiglia di utilizzare il file delle politiche MQIPT invece dei file delle politiche predefiniti descritti in precedenza. Per ulteriori informazioni, consultare **SecurityManagerPolicy** in [MQIPT proprietà globali](#).

La sintassi del file delle politiche è piuttosto complessa e, sebbene sia possibile modificarla utilizzando un editor di testo, di solito è più semplice utilizzare il programma di utilità Policy Tool fornito con Java per apportare eventuali modifiche. Il programma di utilità Policy Tool è disponibile nella directory `$MQIPT_PATH/java/jre/bin` ed è completamente documentato nella documentazione Java.

Un file della politica di esempio (`mqiptSample.policy`) è stato fornito con MQIPT per mostrare quali autorizzazioni devono essere impostate per l'esecuzione di MQIPT.

È necessario modificare il file della politica di esempio in modo che corrisponda alla configurazione. In particolare, tenere presente che la directory home di MQIPT che contiene il file di configurazione `mqipt.conf` potrebbe non essere uguale alla directory di installazione di MQIPT, quindi prestare attenzione a specificare le directory corrette durante la configurazione delle voci **FilePermission** nella politica di sicurezza.

È necessario modificare le seguenti voci:

- La voce **java.io.FilePermission** che concede l'accesso in lettura e scrittura alla directory `errors`. Il percorso file in questa voce deve fare riferimento alla directory home di MQIPT, poiché è qui che si trova la directory `errors`. MQIPT crea i file FFST Failure Data Capture (AMQ\*.FDC) e i file di traccia (AMQ\*.TRC\*) nella directory `errors`. È necessario assicurarsi che MQIPT disponga dell'autorizzazione per creare file di traccia e FFST nella directory `errors`, in modo che sia possibile la risoluzione dei problemi.
- La voce **java.io.FilePermission** che concede l'accesso in lettura e scrittura alla directory `logs`. Il percorso file in questa voce deve fare riferimento alla directory home di MQIPT, poiché è qui che si trova la directory `logs`. MQIPT crea i file di log di connessione (mqipt\*.log) nella directory `logs` se la proprietà globale **ConnectionLog** è abilitata.
- Le voci **java.io.FilePermission** che concedono l'accesso `read` e `execute` a qualsiasi directory nella directory di installazione di MQIPT, come le directory `bin`, `exits`, `libe` `ssl`. I percorsi dei file in queste voci devono essere modificati per fare riferimento alla directory di installazione di MQIPT. Alcune di queste voci possono essere omesse se non sono richieste.
- Le voci **java.net.SocketPermission** devono essere modificate per controllare le connessioni in ciascun instradamento MQIPT in ascolto. Le autorizzazioni `listen` e `accept` sono richieste per la porta del listener e l'indirizzo del listener per ogni instradamento MQIPT.
- Le voci **java.net.SocketPermission** devono essere modificate per controllare le connessioni al di fuori di ogni instradamento MQIPT. L'autorizzazione `connect` è richiesta per tutte le destinazioni di instradamento, i server proxy o i server LDAP a cui si connette l'instradamento MQIPT. L'autorizzazione `resolve` è richiesta quando si specificano destinazioni utilizzando un nome host invece di un indirizzo IP.

A seconda della configurazione, potrebbe essere necessario aggiungere anche le seguenti voci:

- Una voce **java.io.FilePermission** per concedere l'accesso in lettura al file di configurazione `mqipt.conf` o la directory home MQIPT contenente `mqipt.conf`.
- Una voce **java.io.FilePermission** per concedere l'accesso in lettura al file della politica di sicurezza. Ciò è utile se un aggiornamento MQIPT fa sì che il file della politica di sicurezza venga riletto.
- Alcune voci **java.io.FilePermission** per concedere l'accesso in lettura ai file keyring SSL/TLS e ai file di password keyring. Ciò è richiesto solo quando si utilizza un instradamento con le proprietà **SSLClient** o **SSLServer** abilitate o quando è configurata la porta comandi TLS.
- Alcune voci **java.io.FilePermission** per concedere l'accesso `read` o `execute` a qualsiasi classe di uscita MQIPT. Ciò è richiesto solo quando è abilitata un'uscita MQIPT. Potrebbe essere necessario concedere ulteriori autorizzazioni, se richiesto dall'uscita.

**Nota:** Le voci Windows **java.io.FilePermission** devono utilizzare due caratteri barra rovesciata (\\) per ogni barra rovesciata nel percorso. Ciò è dovuto al fatto che una singola barra rovesciata viene utilizzata come carattere escape.

Il file di esempio presuppone che MQIPT sia stato installato su un sistema Windows in `C:\Program Files\IBM\MQ Internet Pass-Thru`. Si presuppone inoltre che la directory home di MQIPT (l'ubicazione del file `mqipt.conf`) sia uguale alla directory di installazione di MQIPT.

Se MQIPT è stato installato in un'altra posizione, è necessario modificare la directory nella definizione **codeBase** per fare riferimento alla directory di installazione di MQIPT. Fare attenzione a includere il prefisso corretto (`file:/`) e il suffisso file corretto (`/lib/com.ibm.mq.ipt.jar`). Sui sistemi AIX and Linux, un URL **codeBase** tipico potrebbe essere `file:/opt/mqipt/lib/com.ibm.mq.ipt.jar`, supponendo che MQIPT sia installato in `/opt/mqipt`.

Le autorizzazioni sono generalmente definite con tre attributi. Per controllare le connessioni socket, i loro valori sono:

#### autorizzazione classe

```
java.net.SocketPermission
```

## nome da controllare

Questo è costituito dal formato `hostname:port`, in cui ogni componente del nome può essere specificato da un carattere jolly. Il nome host può essere un nome dominio o un indirizzo IP. La posizione più a sinistra del nome host può essere specificata da un asterisco (\*). Ad esempio, `harry.company1.com` corrisponderebbe a ciascuna di queste stringhe:

- `harry`
- `harry.company1.com`
- `*.company1.com`
- `*`
- `198.51.100.123` (supponendo che questo sia l'indirizzo IP di `harry.company1.com`)

Il componente porta del nome può essere specificato come un indirizzo porta singolo o un intervallo di indirizzi porta, ad esempio:

### **1414**

solo porta 1414

### **1414-**

tutti gli indirizzi di porta maggiori o uguali a 1414

### **-1414**

tutti gli indirizzi di porta minori o uguali a 1414

### **1-1414**

tutti gli indirizzi di porta compresi tra 1 e 1414, inclusi

## Azione consentita

Le azioni utilizzate da `java.net.SocketPermission` sono:

### **accetta**

Consenti l'accettazione delle connessioni dalla destinazione specificata

### **connect**

Consenti connessioni alla destinazione specificata

### **restare in ascolto**

Consenti all'applicazione di ascoltare le richieste di connessione sulla porta o sulle porte specificate

### **risolvere**

Consenti DNS da utilizzare per risolvere i nomi dominio in indirizzi IP

Il controllo di Java security manager può essere effettuato anche tramite le proprietà di sistema `java.security.manager` e `java.security.policy` Java, ma si consiglia di utilizzare le proprietà **SecurityManager** e **SecurityManagerPolicy** per il controllo MQIPT.

Per includere informazioni diagnostiche nella traccia e nei record FFST, MQIPT deve accedere a determinate proprietà di sistema e variabili di ambiente MQIPT. È necessario includere sempre le seguenti proprietà nella politica di sicurezza Java:

```
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.runtime.version", "read";
permission java.util.PropertyPermission "java.vm.info", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.lang.RuntimePermission "getenv.MQIPT_PATH";
permission java.lang.RuntimePermission "getStackTrace";
permission javax.management.MBeanServerPermission "createMBeanServer";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#[com.ibm.mq.ipt:type=IPTManager]", "registerMBean";
permission javax.management.MBeanPermission "com.ibm.mq.ipt.IPTManager#[com.ibm.mq.ipt:type=IPTManager]", "unregisterMBean";
permission javax.management.MBeanTrustPermission "register";
```

Se non si includono tutte queste proprietà, MQIPT non funzionerà correttamente e la diagnostica del problema sarà compromessa.

## Uscite di sicurezza in MQIPT

Utilizzare un'uscita di sicurezza per controllare l'accesso a una destinazione, come definita dalla proprietà di instradamento **Destination**. L'uscita di sicurezza viene richiamata nel punto in cui MQIPT riceve una richiesta di connessione da un client, ma prima di effettuare la connessione alla destinazione di destinazione.

In base alle proprietà di connessione iniziali, l'uscita di sicurezza decide se è possibile completare la connessione.

Quando viene avviato un instradamento, l'uscita di sicurezza viene richiamata per inicializzarsi e per prepararsi a elaborare una richiesta di collegamento. Il processo di inicializzazione deve essere utilizzato per caricare tutti i dati utente e preparare questi dati per un accesso rapido e semplice, riducendo così il tempo necessario per elaborare una richiesta di connessione.

Ogni instradamento può avere la propria uscita di sicurezza.

- La proprietà **SecurityExit** viene utilizzata per abilitare / disabilitare l'uscita di sicurezza definita dall'utente.
- La propriet ... **SecurityExitName** viene utilizzata per definire il nome classe dell'exit di sicurezza definita dall'utente.
- La proprietà **SecurityExitPath** viene utilizzata per definire il nome della directory contenente il file di classe. Se questa proprietà non è impostata, si presuppone che il file di classe si trovi nella sottodirectory delle uscite. **SecurityExitPath** può anche definire il nome di un file JAR contenente l'uscita di sicurezza definita dall'utente.
- La proprietà **SecurityExitTimeout** viene utilizzata da MQIPT per determinare per quanto tempo deve attendere una risposta dall'uscita di sicurezza durante la convalida di una richiesta di connessione.

Consultare [MQIPT proprietà di instradamento](#) per dettagli sulle proprietà dell'uscita di sicurezza.

MQIPT utilizza la classe `SecurityExit` per richiamare un'uscita di sicurezza definita dall'utente. Questa classe deve essere estesa dall'uscita di sicurezza definita dall'utente e la maggior parte dei suoi metodi deve essere sovrascritta per fornire la funzionalità richiesta. Un oggetto `SecurityExitResponse` viene utilizzato per restituire i dati a MQIPT e questi dati vengono usati da MQIPT per decidere se la richiesta di connessione deve essere accettata o rifiutata. L'oggetto `SecurityExitResponse` può contenere anche una nuova destinazione e un indirizzo di porta di destinazione, utilizzati per sovrascrivere l'instradamento definito dalle proprietà dell'uscita di sicurezza.

Vengono fornite tre uscite di sicurezza di esempio per mostrare come è possibile implementare un'uscita di sicurezza.

- `SampleSecurityExit` mostra come controllare l'accesso a un gestore code IBM MQ, in base al nome del canale IBM MQ. Consente solo una connessione con un nome canale che inizia con la stringa "MQIPT." Per ulteriori informazioni, consultare [Utilizzo di un'uscita di sicurezza](#).
- `SampleRoutingExit` consente l'instradamento dinamico delle richieste di connessioni client a un pool di server IBM MQ definiti, ciascun server che ospita un gestore code con lo stesso nome e gli stessi attributi. L'esempio include un file di configurazione che contiene un elenco di nomi server. Per ulteriori informazioni, consultare [Instradamento delle richieste di connessione client ai server del gestore code IBM MQ utilizzando le uscite di sicurezza](#).
- `SampleOneRouteExit` consente l'instradamento dinamico a un gestore code IBM MQ derivato dal nome del canale IBM MQ utilizzato nella richiesta di connessione. L'esempio include un file di configurazione che contiene una mappa di nomi gestore code e nomi server. Per ulteriori informazioni, consultare [Instradamento dinamico delle richieste di connessione client](#).

**Nota:** MQIPT viene eseguito in una singola JVM, in modo che un'uscita di sicurezza definita dall'utente possa compromettere il normale funzionamento di MQIPT in uno dei seguenti modi:

- Influisce sulle risorse di sistema

- Genera colli di bottiglia
- Prestazioni ridotte

È necessario verificare ampiamente gli effetti dell'uscita di sicurezza prima di implementarla in un ambiente di produzione.

## La classe `com.ibm.mq.ipc.exit.SecurityExit` in MQIPT

Questa classe e i relativi metodi pubblici devono essere estesi dall'uscita di sicurezza definita dall'utente per accedere ad alcuni dati comuni e consentire l'inizializzazione di MQIPT .

Prima che ogni metodo venga richiamato da MQIPT, alcune proprietà saranno rese disponibili per l'utilizzo da parte del metodo. I loro valori possono essere richiamati utilizzando i metodi `get` appropriati definiti in questa classe.

### Metodi

#### **init** `int` pubblico (**IPTrace**)

Sono disponibili le seguenti proprietà:

- porta del listener
- destinazione
- porta di destinazione
- versione

Il metodo `init` viene richiamato da MQIPT quando viene avviato un instradamento. Al ritorno da questo metodo, l'exit di protezione deve essere pronta per convalidare una richiesta di collegamento. I codici di ritorno validi sono `ExitRc.OK` o `ExitRc.INIT_ERROR`.

#### **aggiornamento** `int` pubblico (**IPTrace**)

Sono disponibili le seguenti proprietà:

- porta del listener
- destinazione
- porta di destinazione

Il metodo `refresh` viene richiamato da MQIPT quando la configurazione MQIPT viene aggiornata. Questa azione di solito viene eseguita quando una proprietà è stata modificata nel file di configurazione. MQIPT ricarica tutte le proprietà dal file di configurazione per determinare quali proprietà sono state modificate e se è necessario riavviare un instradamento.

Questo metodo deve eseguire un ricaricamento di tutti i dati esterni che utilizza, ossia i dati caricati dal metodo `init` . I codici di ritorno validi sono `ExitRc.OK` o `ExitRc.REFRESH_ERROR`.

#### **public void close** (**IPTrace**)

Sono disponibili le seguenti proprietà:

- porta del listener
- destinazione
- porta di destinazione

Il metodo `close` viene richiamato da MQIPT quando viene arrestato. Questo metodo dovrebbe liberare tutte le risorse di sistema che l'uscita ha acquisito durante il suo funzionamento. MQIPT attende il completamento di questo metodo prima di chiudere.

Questo metodo viene richiamato anche se un'uscita di sicurezza è stata precedentemente abilitata, ma ora è stata disabilitata nel file di configurazione.

#### **convalida della risposta** `SecurityExit` pubblica (**IPTrace**)

Sono disponibili le seguenti proprietà:

- porta del listener

- destinazione
- porta di destinazione
- tempo massimo
- Indirizzo IP client
- indirizzo porta client
- nome canale
- Nome gestore code

Il metodo `validate` viene richiamato da MQIPT quando riceve una richiesta di connessione da convalidare. Il nome del canale e il nome del gestore code non saranno disponibili se la proprietà **SSLProxyMode** è stata abilitata, poiché questa funzione viene utilizzata solo per eseguire il tunneling dei dati TLS e pertanto i dati generalmente ottenuti dal flusso di dati iniziale non sono leggibili.

L'uscita di protezione deve restituire un oggetto `SecurityExitResponse`, contenente le seguenti informazioni:

- codice motivo (deve essere impostato)
- nuovo indirizzo di destinazione (facoltativo)
- nuovo indirizzo porta listener di destinazione (facoltativo)
- Messaggio (opzionale)

Il codice di errore determina se la connessione viene accettata o rifiutata da MQIPT. I campi `newDestination` e `newDestinationPort` possono essere facoltativamente impostati per definire un nuovo gestore code di destinazione. Se non si impostano queste proprietà, vengono utilizzate le proprietà di instradamento **Destination** e **DestinationPort** definite nel file di configurazione. Qualsiasi messaggio restituito viene aggiunto alla voce del file di log della connessione.

I seguenti metodi sono supportati per ottenere i valori di proprietà di configurazione MQIPT :

**public int getListenerPorta ()**

richiama la porta del listener di instradamento - come definito dalla proprietà **ListenerPort**

**public String getDestination()**

richiama l'indirizzo di destinazione - come definito dalla proprietà **Destination**

**public int getDestinationPort ()**

richiama l'indirizzo della porta del listener di destinazione - come definito dalla proprietà **DestinationPort**

**public String getClientIPAddress ()**

richiama l'indirizzo IP del client che effettua la richiesta di collegamento

**public int getClientPortAddress()**

richiama l'indirizzo porta utilizzato dal client che effettua la richiesta di connessione

**public int getTimeout()**

richiama il valore di timeout. MQIPT attenderà che l'uscita di sicurezza convalidi una richiesta - come definito dalla proprietà **SecurityExitTimeout**

**public int getConnThreadID()**

richiama l'ID thread di collegamento che gestisce la richiesta di collegamento, utile per il debug

**public String getChannelNome ()**

richiama il nome canale IBM MQ utilizzato nella richiesta di collegamento

**public String getQMName()**

richiama il nome del gestore code IBM MQ utilizzato nella richiesta di connessione

**Booleano pubblico getTimedout()**

può essere utilizzato dall'uscita di sicurezza per stabilire se il timeout è scaduto

## La classe com.ibm.mq.ipt.exit.SecurityExitResponse in MQIPT

Questa classe viene utilizzata per passare una risposta a MQIPT da un'exit di sicurezza definita dall'utente e viene utilizzata per stabilire se la richiesta di connessione deve essere accettata o rifiutata.

Gli oggetti di questo tipo vengono creati solo nel metodo di convalida (consultare [“La classe com.ibm.mq.ipt.exit.SecurityExit in MQIPT” a pagina 1053](#)). Esistono costruttori di convenienza per la creazione di questi oggetti ed esistono metodi per ogni proprietà. Consultare le uscite di sicurezza di esempio per ulteriori informazioni.

La creazione di un oggetto risposta SecurityExitpredefinito rifiuta la richiesta di connessione.

### Costruttori

- **public SecurityExitResponse (String dest, int destPort, int rc, String msg)**

dove:

- dest è la nuova destinazione
- destPort è il nuovo indirizzo porta di destinazione
- rc è il codice di errore
- msg è un messaggio che verrà aggiunto alla voce del log di connessione

- **public SecurityExitResponse (String dest, int destPort, int rc)**
- **public SecurityExitRisposta (int rc, String msg)**
- **SecurityExitRisposta (int rc)**

### Metodi

#### **public void setDestination(String dest)**

imposta un indirizzo di destinazione nuovo per la richiesta di collegamento

#### **public void setDestinationLa porta (int port) genera IPTException**

imposta un nuovo indirizzo di porta del listener di destinazione per la richiesta di connessione - genera un'eccezione IPTException per un indirizzo di porta non valido

#### **public void setMessage(String msg)**

aggiunge un messaggio al record di log di connessione

#### **public void setReasonCodice (int rc)**

imposta il codice motivo per la richiesta di collegamento.

## Codici di ritorno dell'uscita di sicurezza in MQIPT

I codici di ritorno che MQIPT riconosce quando richiama un'uscita di sicurezza in diverse situazioni.

I seguenti codici di ritorno vengono riconosciuti da MQIPT quando si richiama un'uscita di sicurezza nelle seguenti situazioni:

Codice di ritorno	Descrizione	inizializzare	Convalida	Aggiornare
ExitRc.OK	Richiesta completata correttamente.	sì	sì	sì
ExitRc.ERRORE DI INIZIALIZZAZIONE	Richiesta di inizializzazione non riuscita, l'instradamento verrà disabilitato.	sì		
ExitRc.ERRORE_RISPEDIZIONE	Richiesta di aggiornamento non riuscita.			sì
ExitRc.NON_AUTORIZZATO	Processo di convalida non riuscito, richiesta di connessione rifiutata.		sì	

Codice di ritorno	Descrizione	inizializzare	Convalida	Aggiornare
ExitRc.DISABILITAZIONE_SSL	Richiesta di convalida riuscita, la connessione alla destinazione non utilizzerà SSL o TLS.		sì	

## Controllo numero porta in MQIPT

Quando si utilizza MQIPT, è possibile limitare l'intervallo del numero di porta locale utilizzato quando si effettua una connessione in uscita.

Impostare la proprietà **OutgoingPort** sull'instradamento per specificare il numero di porta locale iniziale e impostare **MaxConnectionThreads** per specificare il numero di porte da utilizzare. Ad esempio, se si imposta **OutgoingPort** su 1600 e **MaxConnectionThreads** su 20, l'intervallo dei numeri di porta locali per tale instradamento è 1600 - 1619.

È responsabilità dell'amministratore di MQIPT assicurarsi che non vi siano conflitti di numeri di porta tra gli instradamenti.

Se **OutgoingPort** non è definito, un valore predefinito di 0 indica che per ogni connessione viene utilizzato un numero di porta assegnato dal sistema.

Quando si utilizza HTTP, il numero di porte in uscita è doppio rispetto a quando non si utilizza HTTP. Nell'esempio precedente, se l'instradamento utilizzava HTTP, l'intervallo di numeri sarebbe 1600 - 1639.

Per ulteriori informazioni, consultare [Assegnazione di numeri di porte](#).

## Sistemi multihomed

Quando si usa un sistema multihomed, è possibile specificare a quale indirizzo IP si collegherà una connessione in uscita utilizzando la proprietà **LocalAddress**. I nomi host non sono supportati su questa proprietà.

## Crittografia delle password memorizzate in MQIPT

La configurazione di MQIPT potrebbe includere le password per accedere a varie risorse, nonché la password per accedere a MQIPT utilizzando la porta comandi. Tutte queste password devono essere protette mediante crittografia.

### Informazioni su questa attività

Tutte le password memorizzate per l'utilizzo da parte di MQIPT devono essere protette codificando la password con il comando **mqiptPW**. Le password codificate vengono memorizzate come valori di proprietà nel file di configurazione **mqipt.conf**. MQIPT è in grado di distinguere tra password codificate, password in testo semplice e nomi file nei valori delle proprietà. È necessario crittografare tutte le password memorizzate per l'utilizzo da parte di MQIPT in questo modo in quanto è il metodo di protezione più sicuro.

Per migliorare la protezione delle password key ring, codificare nuovamente le password key ring precedentemente codificate, utilizzando il metodo di protezione più recente.

**Nota:** La propriet ... **SSLCommandPortKeyRingPW** nel file di configurazione **mqipt.conf** e la propriet ... **SSLClientCAKeyRingPW** nel file delle propriet ... **mqiptAdmin** non possono fare riferimento ai file delle password. I valori di queste proprietà devono essere impostati sull'output della stringa della password codificata dal comando **mqiptPW**.

Se nella configurazione di MQIPT è presente un testo semplice o una password debolmente protetta, viene emesso un messaggio di avviso quando viene avviato MQIPT o quando viene avviato un instradamento.



Utilizzare questa procedura per codificare una parola d'ordine da memorizzare per l'utilizzo da parte di MQIPT utilizzando il metodo di protezione più recente.

## Procedura

1. Opzionale: Creare un file contenente la chiave di codifica della password, se non si dispone già di una chiave.

MQIPT utilizza una chiave di codifica per codificare le password. È possibile specificare la propria chiave di codifica in un file. Il file deve contenere almeno un carattere e solo una riga di testo.

La stessa chiave di codifica della parola d'ordine viene utilizzata per codificare e decodificare tutte le parole d'ordine memorizzate per un'istanza di MQIPT. Pertanto, è necessario solo un singolo file di chiavi di codifica della parola d'ordine per ogni installazione di MQIPT .

È possibile utilizzare una chiave di codifica password diversa per codificare le password memorizzate nel file delle proprietà **mqiptAdmin** rispetto alla chiave di codifica utilizzata per codificare le password nella configurazione di MQIPT .

Se si intende eseguire MQIPT come un servizio avviato automaticamente, è necessario creare il file della chiave di codifica della password con il nome predefinito `mqipt_cred.key` e posizionarlo nella directory home MQIPT .

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.

**Nota:** È necessario assicurarsi che le autorizzazioni file appropriate siano impostate sul file della chiave di codifica della parola d'ordine 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.

2. Codificare la password utilizzando il comando **mqiptPW**.

La sintassi del comando **mqiptPW** è descritta in [mqiptPW \(encrypt stored password\)](#).

Se è stato creato un file di chiavi di codifica della password nel passo “1” a pagina 1057, specificare il nome file utilizzando il parametro **-sf** su **mqiptPW**. Ad esempio, il seguente comando può essere immesso per codificare una parola d'ordine utilizzando la chiave di codifica nel file specificato dal parametro **-sf** :

```
mqiptPW -sf /opt/mqipt/mqipt_password.key
```

3. Immettere la password da codificare quando richiesto.

La password codificata verrà emessa da **mqiptPW**.

4. Copiare la password codificata nella proprietà appropriata nel file di configurazione `mqipt.conf` o nel file delle proprietà **mqiptAdmin**.

Ad esempio, la seguente riga specifica una parola d'ordine codificata per la parola d'ordine di accesso MQIPT :

```
AccessPW=<mqiptPW>1!QL+2Jvj/tigKK1D7Nz80qw==!AMDBef0UrmPf5i10uqV5MA==
```

5. Avviare MQIPT. Se è stato creato un file di chiavi di codifica della password nel passo “1” a pagina 1057 con un nome diverso da quello predefinito, specificare il nome del file di chiavi di codifica all'inizio di MQIPT.

È possibile specificare il nome del file della chiave di codifica password utilizzando il parametro **-sf** quando si avvia MQIPT. Ad esempio, immettere il seguente comando per avviare MQIPT utilizzando la chiave di crittografia nel file specificato dal parametro **-sf** :

```
mqipt /opt/mqipt -sf /opt/mqipt/mqipt_password.key
```

Per informazioni su altri metodi per specificare il nome file della chiave di codifica della password all'avvio di MQIPT, consultare [Specifica della chiave di codifica della password](#).

È possibile specificare il nome del file di chiavi di codifica della password per il comando **mqiPTAdmin** utilizzando la proprietà **PasswordProtectionKeyFile** nel file di proprietà **mqiPTAdmin**.

## Altre considerazioni sulla sicurezza per MQIPT

MQIPT dispone di diverse funzioni aggiuntive che consentono a un designer di creare una soluzione sicura.

- Se ci sono molti client in una rete interna che tentano di effettuare connessioni in uscita, possono tutti passare attraverso un MQIPT che si trova all'interno del firewall. L'amministratore del firewall deve quindi concedere l'accesso esterno solo al computer MQIPT.
- MQIPT può connettersi solo ai gestori code per cui è stato esplicitamente configurato nel relativo file di configurazione, a meno che MQIPT non agisca come un proxy SOCKS o utilizzi un'uscita di sicurezza.
- MQIPT verifica che i messaggi ricevuti e trasmessi siano validi e conformi al protocollo IBM MQ. Ciò consente di evitare che MQIPT venga utilizzato per attacchi di sicurezza all'esterno del protocollo IBM MQ. Se MQIPT agisce come un proxy SSL/TLS, quando tutti i dati e i protocolli IBM MQ sono stati codificati, MQIPT può garantire solo l'handshake SSL/TLS iniziale. In questa situazione, utilizzare [Java security manager](#).
- MQIPT consente alle uscite canale di eseguire i propri protocolli di sicurezza end - to - end.
- È possibile limitare il numero totale di connessioni in entrata impostando la proprietà `MaxConnectionThreads`. Ciò consente di proteggere un gestore code interno vulnerabile dagli attacchi DoS (denial of service).

## File di configurazione

È necessario proteggere il file di configurazione MQIPT, `mqiPT.conf`, dalla lettura da parte di utenti non autorizzati perché potrebbe contenere informazioni sensibili, come la password **AccessPW** che controlla l'accesso amministrativo remoto a MQIPT. Proteggere tutte le password specificate nel file di configurazione seguendo la procedura in [“Crittografia delle password memorizzate in MQIPT”](#) a pagina 1056. Inoltre, verificare che `mqiPT.conf` sia protetto da modifiche non autorizzate. Impostare le autorizzazioni del file del sistema operativo per `mqiPT.conf` in modo che solo l'account utente che esegue MQIPT possa leggere o aggiornare il file.

## Porta comandi

Le porte di comando MQIPT accettano i comandi di gestione emessi sulla rete per un'istanza remota di MQIPT dal comando **mqiPTAdmin**.

MQIPT può essere configurato con una porta comandi non protetta e una porta comandi protetta con TLS. Le connessioni alla porta comandi non protetta non sono codificate.

**Nota:** I dati inviati tramite la rete alla porta comandi non protetta, inclusa la password di accesso MQIPT, possono essere visibili ad altri utenti sulla rete.

È necessario considerare se è necessario abilitare una porta comandi e valutare i rischi di consentire la gestione remota di MQIPT, prima di abilitare la porta comandi TLS o non protetta. Il comando **mqiPTAdmin** può gestire le istanze locali di MQIPT in esecuzione con lo stesso utente del comando **mqiPTAdmin** senza utilizzare una porta comandi. Pertanto, potrebbe non essere necessario abilitare una porta comandi per amministrare le istanze locali di MQIPT.

Se la porta comandi non protetta o TLS è abilitata, è necessario impedire l'accesso non autorizzato alla porta comandi. Ad esempio, è necessario considerare questi punti quando si protegge l'accesso alla porta comandi:

- Utilizzare un firewall per limitare la serie di computer che possono connettersi alla porta comandi MQIPT.
- Abilitare l'autenticazione sulle porte comandi utilizzando le proprietà **AccessPW** e **RemoteCommandAuthentication**. Per ulteriori informazioni sull'abilitazione dell'autenticazione della porta comandi, consultare [Autenticazione della porta comandi](#).

- Disabilitare l'arresto remoto con la proprietà **RemoteShutdown** .
- Utilizzare le proprietà **CommandPortListenerAddress** e **SSLCommandPortListenerAddress** per configurare le porte dei comandi in ascolto su una specifica interfaccia di rete.

Per ulteriori informazioni sull'utilizzo del comando **mqiptAdmin** per gestire MQIPT, consultare [Amministrazione MQIPT utilizzando la riga comandi](#).

## Log di connessione in MQIPT

MQIPT fornisce una funzione di log di connessione che contiene elenchi di tutti i tentativi di connessione riusciti e non riusciti.

Una voce viene scritta nel log di connessione per ogni collegamento ricevuto o effettuato da un instradamento MQIPT e per ogni comando di gestione ricevuto da MQIPT. Il log di connessione viene controllato utilizzando le proprietà **ConnectionLog** e **MaxLogFileSize** . Per ulteriori informazioni, consultare [MQIPT proprietà globali](#) .

Ogni volta che MQIPT viene avviato, viene creato un nuovo log di connessione. Per l'identificazione, il nomefile include la data / ora corrente, ad esempio:

```
mqiptYYYYMMDDHHmmSS.log
```

dove

YYYY è l'anno  
MM è il mese  
DD è il giorno  
HH sono le ore  
mm è il minuto  
SS sono i secondi

Quando un log di connessione raggiunge la dimensione massima determinata dalla proprietà **MaxLogFileSize** , viene creato un file di backup, `mqipt001.log`. Vengono conservati un numero massimo di due file di backup (`mqipt001.log` e `mqipt002.log`).

Una voce nel log di connessione rappresenta ogni parte di una richiesta di collegamento. Una richiesta di connessione ricevuta da MQIPT e la nuova connessione risultante che MQIPT effettua all'indirizzo di destinazione viene visualizzata come due voci di log e, successivamente, due ulteriori voci al termine di ogni connessione.

Di seguito è riportato il log di connessione per una richiesta di connessione riuscita:

```
Wed May 15 13:13:51 BST 2013 conn accept 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:51 BST 2013 conn conn 127.0.0.1(3843) localhost(3500) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close 127.0.0.1(3843) localhost(3500) OK 5000-0
```

Di seguito è riportato un log di connessione per una richiesta di collegamento non riuscita:

```
Wed May 15 14:56:40 BST 2013 conn accept 127.0.0.1(4138) 127.0.0.1(7000) OK 7000-0
Wed May 15 14:56:40 BST 2013 conn close 127.0.0.1(4138) 127.0.0.1(7000) ERROR 7000-0
Unrecognized SSL handshake request '54'
```

## Voci di log di connessione

Ogni voce del log di collegamento contiene le informazioni riportate di seguito:

- L'ora in cui la voce è stata creata.
- Il tipo di immissione. Il valore può essere uno dei seguenti:

**admin**

Comando di gestione

**CONN**

Instrada connessione

- L'evento che si è verificato. Il valore può essere uno dei seguenti:

**accetta**

Richiesta di connessione ricevuta

**chiudi**

Connessione chiusa

**CONN**

Richiesta di connessione alla destinazione di instradamento

**dspipt**

Comando MQIPT di visualizzazione ricevuto

**nodata**

Nessun dato ricevuto dal chiamante

**ping**

Richiesta ping ricevuta

**stato**

Comando di visualizzazione stato ricevuto

**rif.**

Comando di aggiornamento ricevuto

**arresta**

Comando di arresto ricevuto

- L'indirizzo di rete di origine e numero di porta. Il valore LOCAL viene visualizzato per i comandi di gestione emessi localmente senza utilizzare la porta comandi.
- L'indirizzo di rete di destinazione e numero di porta. Non viene visualizzato per i comandi di gestione emessi localmente senza utilizzare la porta comandi.
- Il codice di completamento Il valore può essere OK o ERROR.
- L'identificativo del thread MQIPT .
- Un messaggio di errore facoltativo.

## Configurazione di IBM MQ Internet Pass-Thru utilizzando i container

Puoi eseguire IBM MQ Internet Pass-Thru (MQIPT) in un contenitore. L'immagine di base utilizzata dall'immagine contenitore deve utilizzare un sistema operativo Linux supportato.

### Procedura

- Nel repository GitHub del contenitore mq è disponibile un'immagine MQIPT Docker di esempio. Per creare ed eseguire il contenitore, segui le istruzioni in [IBM MQ Internet Pass-Thru su Docker](#).

### Operazioni successive

È possibile visualizzare i contenitori in esecuzione utilizzando il comando **docker ps** . Per visualizzare l'output della console di MQIPT in esecuzione su un contenitore Docker , utilizza il comando **docker logs \${CONTAINER\_ID}** .

## Configurazione delle code di flusso

---

La funzione delle code di flusso consente di avere una copia duplicata di ogni messaggio inserito in una coda, consegnata a una seconda coda. La configurazione delle code di flusso viene eseguita su una coda per coda.

Le code locali e di modello hanno due nuovi attributi relativi alle code di flusso:

### **STREAMQ**

Questo è il nome della coda a cui devono essere consegnati i messaggi trasmessi. Impostare l'attributo **STREAMQ** sul nome di un'altra coda.

Ci sono restrizioni su quali code possono essere configurate per trasmettere i messaggi ad altre code e ci sono restrizioni su quali code possono essere impostate come destinazione per i messaggi in streaming. Per informazioni sulle limitazioni del flusso di messaggi, consultare [Limitazioni della coda di flusso](#).

### **STRMQSO**

Questo è il QoS (quality - of - service) da utilizzare per la consegna di messaggi in streaming.

È possibile impostare l'attributo **STRMQOS** su uno dei seguenti valori:

#### **BESTEF**

Massimo sforzo, che è il valore predefinito.

Il gestore code tenta di consegnare una copia di ciascun messaggio alla coda specificata nell'attributo **STREAMQ**. Se si verifica un problema durante la consegna del messaggio in streaming, ciò non influisce sulla consegna del messaggio originale.

#### **MUSTDUP**

Il gestore code tenta di consegnare una copia di ciascun messaggio alla coda di flusso.

Se si verifica un problema durante la consegna del messaggio trasmesso in streaming, il messaggio originale non viene consegnato alla relativa coda e l'applicazione riceve MQCC\_FAILED insieme ad un codice di errore appropriato.

Per ulteriori dettagli, consultare i comandi [ALTER queues](#), [DEFINE queues](#) e [DISPLAY QUEUE MQSC e Change, Copy, and Create Queue, Inquire Queue e Inquire Queue \(Response\) PCF](#).

Se è richiesta più di una copia di ciascun messaggio, è possibile configurare l'attributo **STREAMQ** in modo che faccia riferimento al nome di una coda alias IBM MQ la cui destinazione fa riferimento a un argomento IBM MQ. Quando un messaggio viene inserito nella coda originale, una copia del messaggio viene pubblicata nell'argomento specificato.

È necessario assicurarsi di disporre di API o di sottoscrizioni gestite per l'oggetto argomento quando ogni sottoscrizione riceve una copia del messaggio. Il messaggio consegnato ai sottoscrittori segue le stesse regole degli altri messaggi di pubblicazione / sottoscrizione. Ad esempio, ogni messaggio ha un nuovo identificativo e i campi di contesto di MQMD sono diversi da quelli del messaggio originale. Per ulteriori informazioni sulle similitudini e sulle differenze tra i messaggi originali e in streaming, consultare [Messaggi in streaming](#).

## **Esempi**

### **Esempio di massimo sforzo**

Nel seguente esempio, una coda locale ORDERS.QUEUE viene modificata per inserire i messaggi di flusso in una seconda coda ANALYTICS.QUEUE. La QoS (quality of service) BESTEF viene utilizzata per garantire che se si verifica un problema di inserimento del messaggio in streaming in ANALYTICS.QUEUE, ad esempio, se ANALYTICS.QUEUE è pieno, il messaggio originale può ancora essere inserito in ORDERS.QUEUE.

Questo tipo di configurazione può essere utilizzato per eseguire l'analisi sugli ordini ricevuti, analizzando i messaggi in streaming, mentre i messaggi originali vengono inseriti nella coda degli ordini e elaborati. Un vantaggio della funzione della coda di flusso è che è possibile lasciare i messaggi in streaming su ANALYTICS.QUEUE in attesa di essere elaborato, senza influire sugli ordini effettivi soddisfatti dall'azienda.

```
DEFINE QLOCAL (ANALYTICS.QUEUE)
```

```
ALTER QLOCAL (ORDERS.QUEUE) STRMQOS (BESTEF) STREAMQ (ANALYTICS.QUEUE)
```

**Nota:** Nell'esempio **STRMQOS** è stato impostato su **BESTEF**, sebbene sia possibile lasciare questo attributo fuori dal comando **ALTER**, poiché **BESTEF** è la qualità predefinita del servizio.

### Esempio da duplicare

In questo esempio, una coda locale **PAYMENTS.QUEUE** viene modificata per inserire copie in flusso di ogni messaggio in un'altra coda locale **AUDIT.QUEUE**. È importante che ogni messaggio inserito nella coda di pagamento venga trasmesso alla coda di verifica, quindi viene utilizzata la QoS (quality of service) **MUSTDUP**.

Se si verifica un problema durante la consegna del messaggio di flusso alla relativa coda, anche il messaggio originale non viene consegnato e l'applicazione riceve un codice motivo e di completamento appropriato. L'applicazione deve ritentare l'inserimento nello stesso modo in cui farebbe se fosse coinvolta una sola coda.

```
DEFINE QLOCAL (AUDIT.QUEUE)
```

```
ALTER QLOCAL (PAYMENTS.QUEUE) STRMQOS (MUSTDUP) STREAMQ (AUDIT.QUEUE)
```

#### Note:

1. Non è necessario che la coda di flusso esista quando si modifica la coda originale. Tuttavia, è importante notare che poiché la qualità del servizio utilizzata è **MUSTDUP**, i tentativi di inserire i messaggi nella coda originale non riescono fino a quando non viene definita la coda di flusso.
2. Quando si utilizza un alias della coda con una destinazione di un oggetto argomento, se non vi sono sottoscrittori, la consegna del messaggio in streaming viene ancora considerata riuscita e il messaggio originale viene consegnato alla relativa coda.
3. Se un messaggio in streaming non può essere consegnato alla sua coda, il gestore code non tenta di consegnarlo alla coda di messaggi non recapitabili. Tuttavia, se un messaggio di flusso viene inviato a una coda remota, quando viaggia su un canale verso un altro gestore code, può essere recapitato a una coda di messaggi non recapitabili seguendo le regole dei messaggi non recapitabili esistenti.

### Configurazione della coda di flusso

Non è necessario eseguire alcuna configurazione aggiuntiva sulla coda di flusso. Riceve i messaggi da tutte le code che lo denominano come una coda di flusso. Tuttavia, potrebbe essere opportuno considerare i valori di attributo configurati nella coda di flusso.

Ad esempio, se la coda originale ha una profondità massima di 100.000 e la coda di flusso ha una profondità massima di 5000, i messaggi di flusso potrebbero andare persi se **STRMQOS** è impostato su **BESTEF** o non riuscire al momento dell'immissione se **STRMQOS** è impostato su **MUSTDUP**, con un errore **MQRC\_Q\_FULL**, anche se la coda originale ha ancora molto spazio.

Considerare quali attributi nella coda di flusso potrebbero dover essere modificati per avere valori appropriati, in base alla modalità di configurazione della coda originale.

#### Concetti correlati

[Code di streaming](#)

## Informazioni particolari

---

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti.

IBM potrebbe non offrire i prodotti, i servizi o le funzioni descritti in questo documento in altri paesi. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti da IBM possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM potrebbe disporre di applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura di tale documento non concede alcuna licenza a tali brevetti. Chi desiderasse ricevere informazioni relative a licenze può rivolgersi per iscritto a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali:** INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe non essere applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche vengono incorporate nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti o modifiche al prodotto/i e/o al programma/i descritti nella pubblicazione in qualsiasi momento e senza preavviso.

Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti solo per consultazione e non rappresenta in alcun modo un'approvazione di tali siti. I materiali reperibili in tali siti Web non fanno parte dei materiali relativi a questo prodotto IBM e l'utilizzo di tali siti è responsabilità dell'utente.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente da IBM e diventeranno esclusiva della stessa.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Corporation  
Coordinatore interoperabilità software, Dipartimento 49XA  
Autostrada 3605 52 N

Rochester, MN 55901  
U.S.A.

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in queste informazioni e tutto il materiale su licenza disponibile per esso sono forniti da IBM in base ai termini dell' IBM Customer Agreement, IBM International Program License Agreement o qualsiasi altro accordo equivalente tra le parti.

Tutti i dati relativi alle prestazioni contenuti in questo documento sono stati determinati in un ambiente controllato. Pertanto, i risultati ottenuti in altri ambienti operativi possono variare in modo significativo. Alcune misurazioni potrebbero essere state fatte su sistemi a livello di sviluppo e non vi è alcuna garanzia che queste misurazioni saranno le stesse sui sistemi generalmente disponibili. Inoltre, alcune misurazioni potrebbero essere state stimate mediante estrapolazione. I risultati quindi possono variare. Gli utenti di questo documento dovrebbero verificare i dati applicabili per il loro ambiente specifico.

Le informazioni relative a prodotti non IBM provengono dai fornitori di tali prodotti, dagli annunci pubblicati o da altre fonti pubblicamente disponibili. IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Tutte le dichiarazioni riguardanti la direzione o l'intento futuro di IBM sono soggette a modifica o ritiro senza preavviso e rappresentano solo scopi e obiettivi.

Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali. Per poterli illustrare nel modo più completo possibile, gli esempi riportano nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi adoperati da imprese realmente esistenti sono una mera coincidenza.

#### LICENZA SUL COPYRIGHT:

Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi.

Se si sta visualizzando queste informazioni in formato elettronico, le fotografie e le illustrazioni a colori potrebbero non apparire.

## Informazioni sull'interfaccia di programmazione

---

Le informazioni sull'interfaccia di programmazione, se fornite, consentono di creare software applicativo da utilizzare con questo programma.

Questo manuale contiene informazioni sulle interfacce di programmazione che consentono al cliente di scrivere programmi per ottenere i servizi di IBM MQ.

Queste informazioni, tuttavia, possono contenere diagnosi, modifica e regolazione delle informazioni. La diagnosi, la modifica e la regolazione delle informazioni vengono fornite per consentire il debug del software applicativo.

**Importante:** Non utilizzare queste informazioni di diagnosi, modifica e ottimizzazione come interfaccia di programmazione poiché sono soggette a modifica.

## Marchi

---

IBM, il logo IBM, ibm.com, sono marchi di IBM Corporation, registrati in molte giurisdizioni nel mondo. Un elenco aggiornato dei marchi IBM è disponibile sul web in "Copyright and trademark



information"www.ibm.com/legal/copytrade.shtml. Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o altre società.

Microsoft e Windows sono marchi di Microsoft Corporation negli Stati Uniti, in altri paesi o entrambi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e/o in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Questo prodotto include il software sviluppato da Eclipse Project (<https://www.eclipse.org/>).

Java e tutti i marchi e i logo Java sono marchi registrati di Oracle e/o di società affiliate.







Numero parte:

(1P) P/N: